Microsoft

Visualizing Large-scale and High-dimensional Data

Jian Tang, Jingzhou Liu, Ming Zhang and Qiaozhu Mei

Microsoft Research

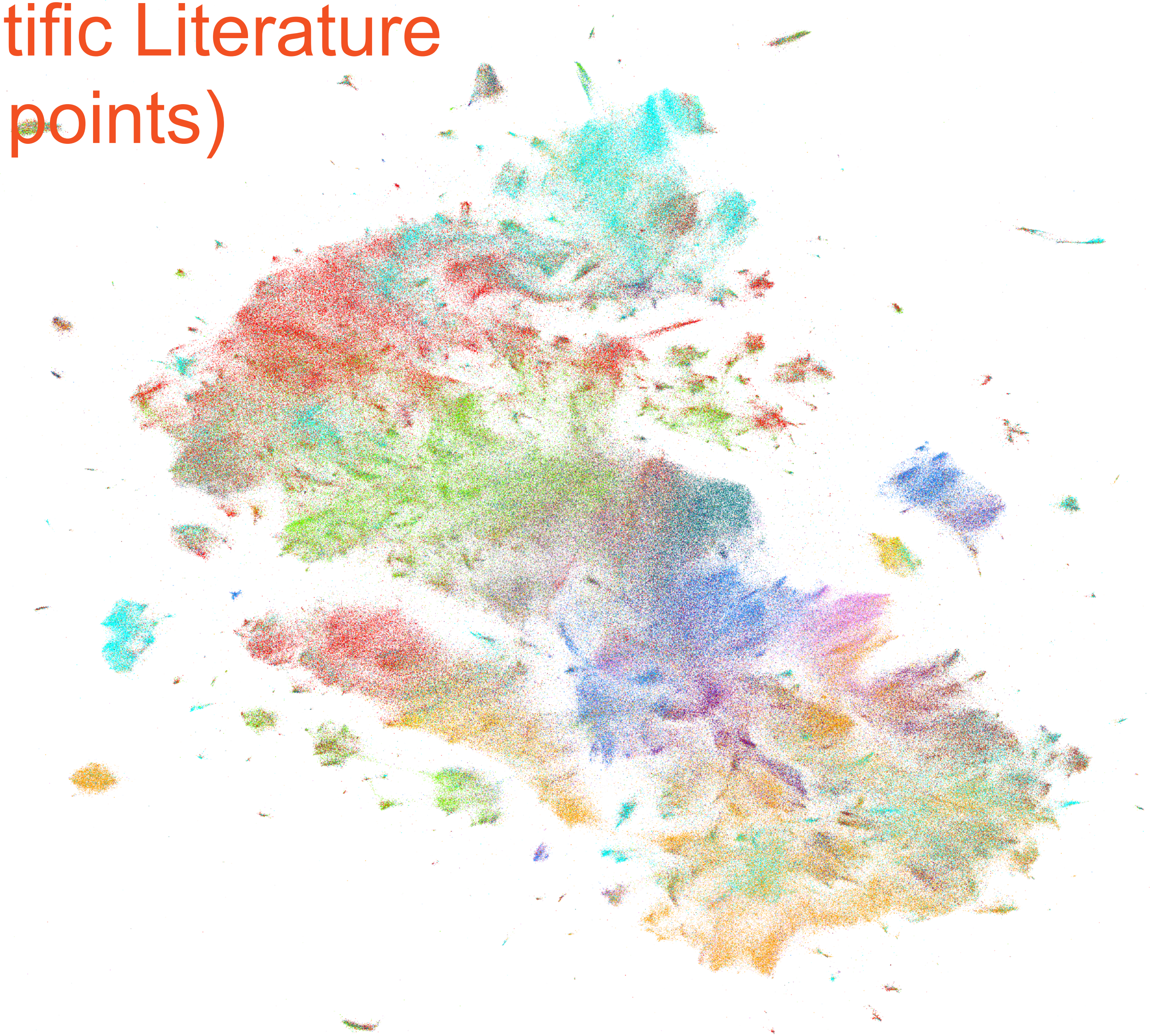
PEKING UNIVERSITY 1898

UNIVERSITY OF MICHIGAN

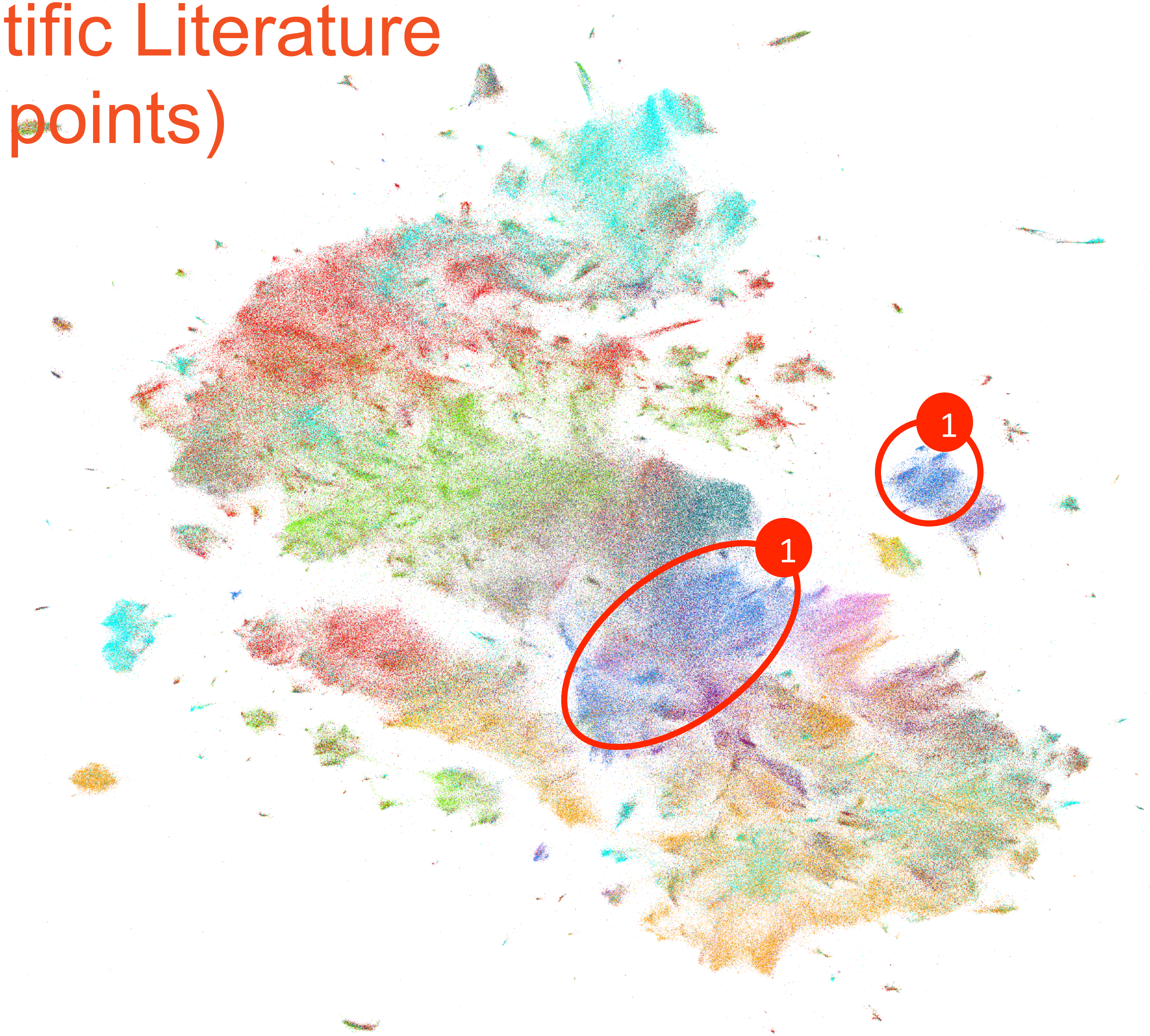
Where is **Computer Science** in the map of all **Sciences**?

Where is **WWW** among all fields of **Computer Science**?

Scientific Literature (10M points)

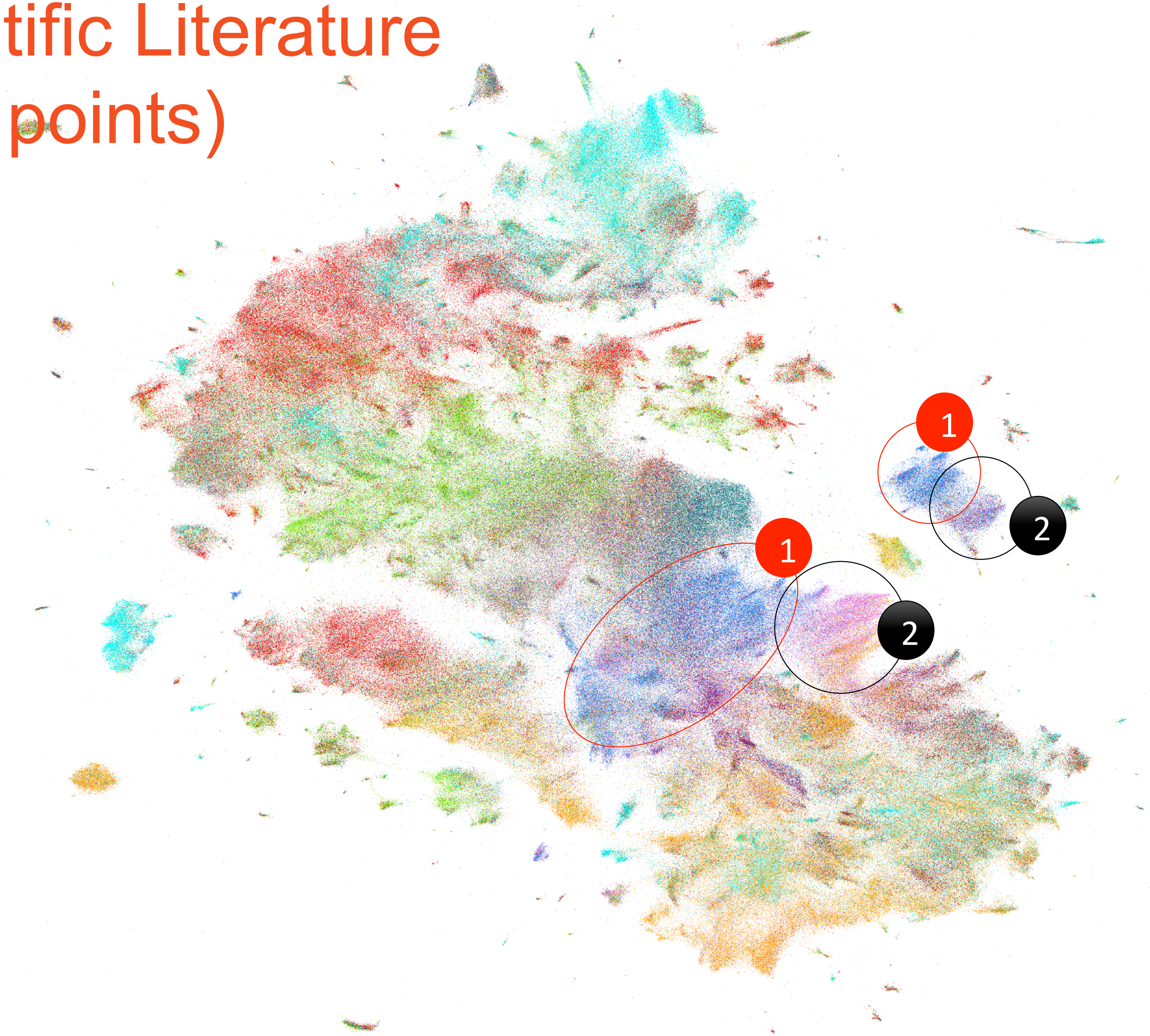


Scientific Literature (10M points)



1 Computer Science

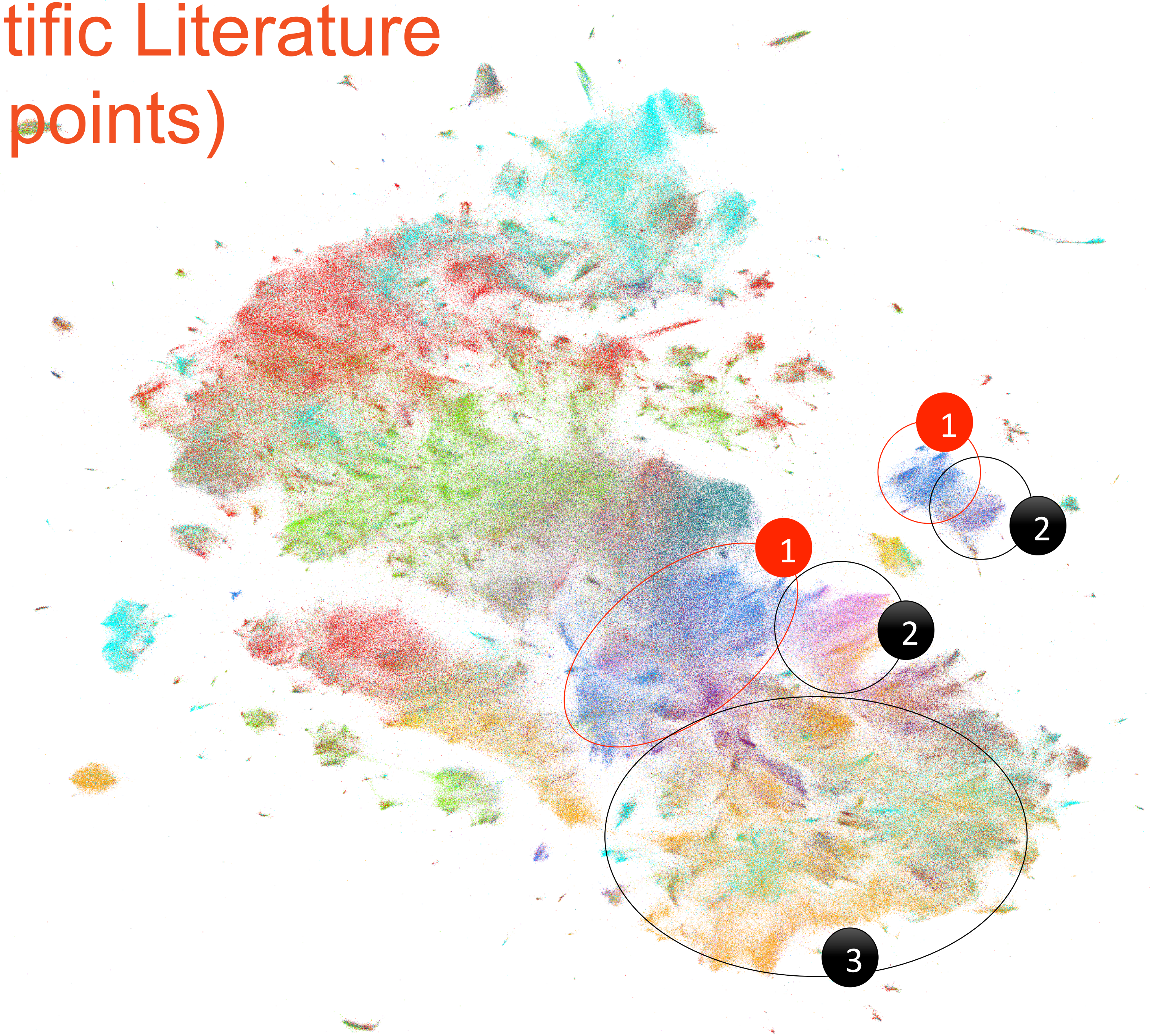
Scientific Literature (10M points)



1 Computer Science

2 Mathematics

Scientific Literature (10M points)

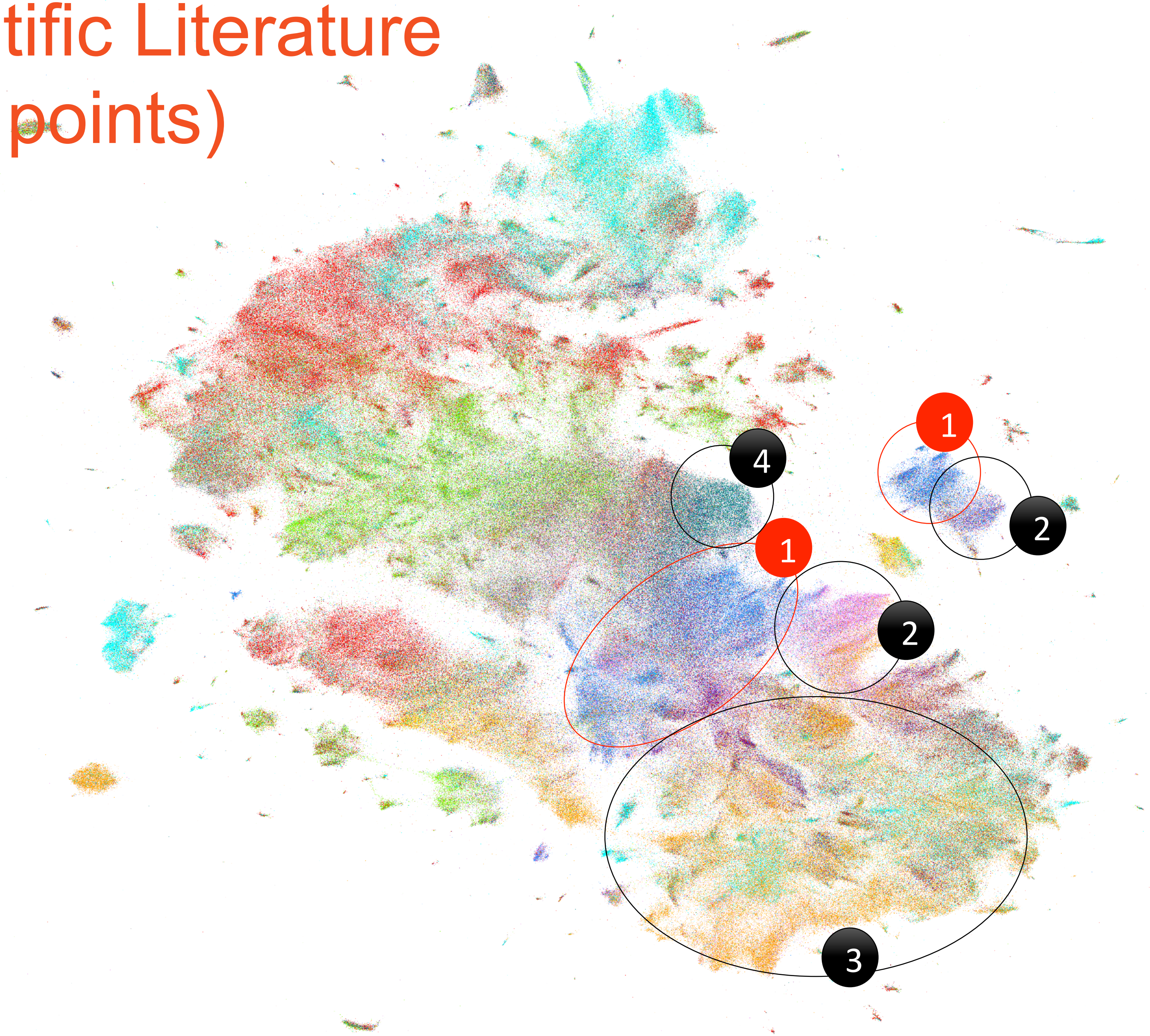


1 Computer Science

2 Mathematics

3 Physics

Scientific Literature (10M points)



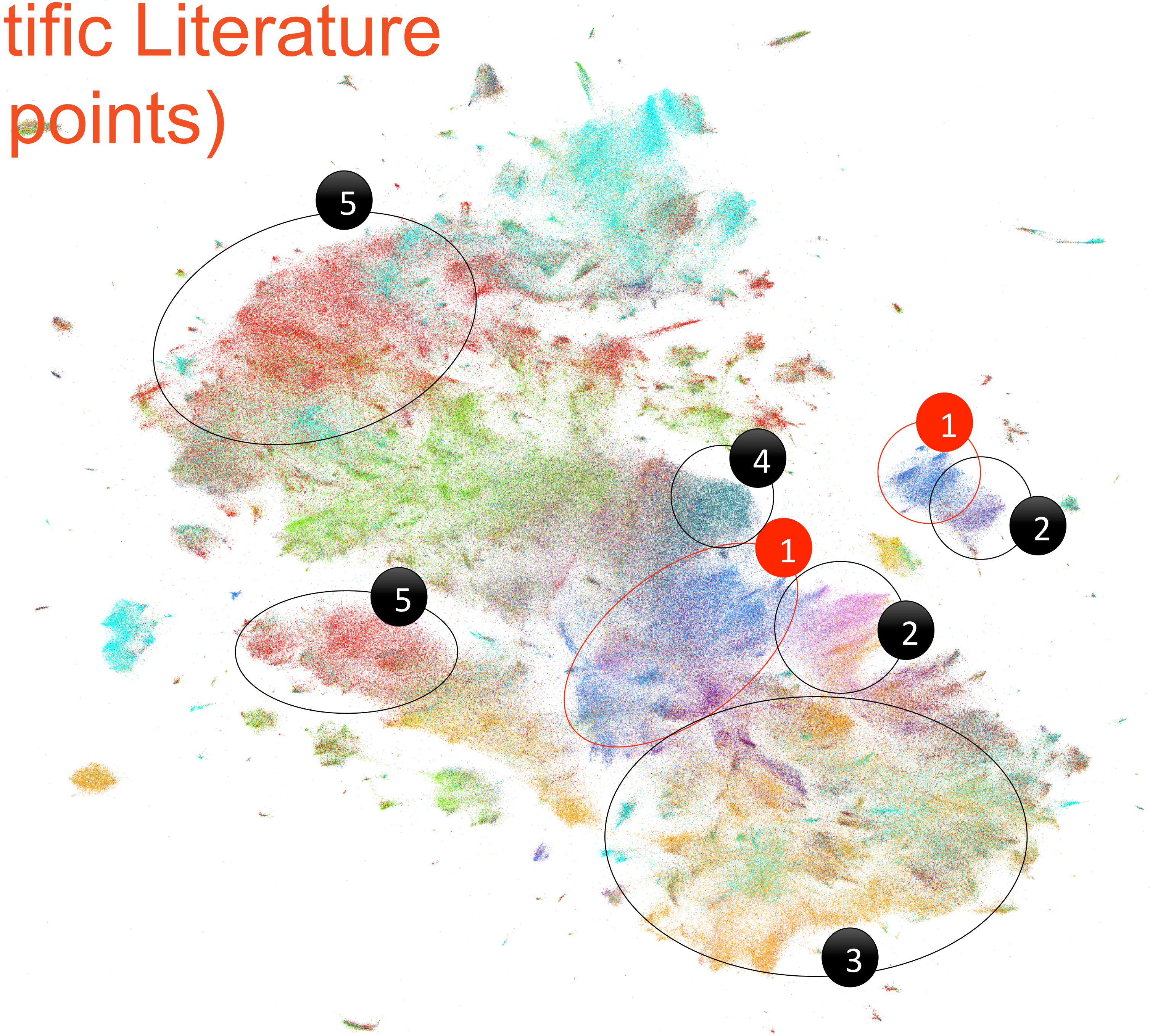
1 Computer Science

2 Mathematics

3 Physics

4 Economics

Scientific Literature (10M points)



1 Computer Science

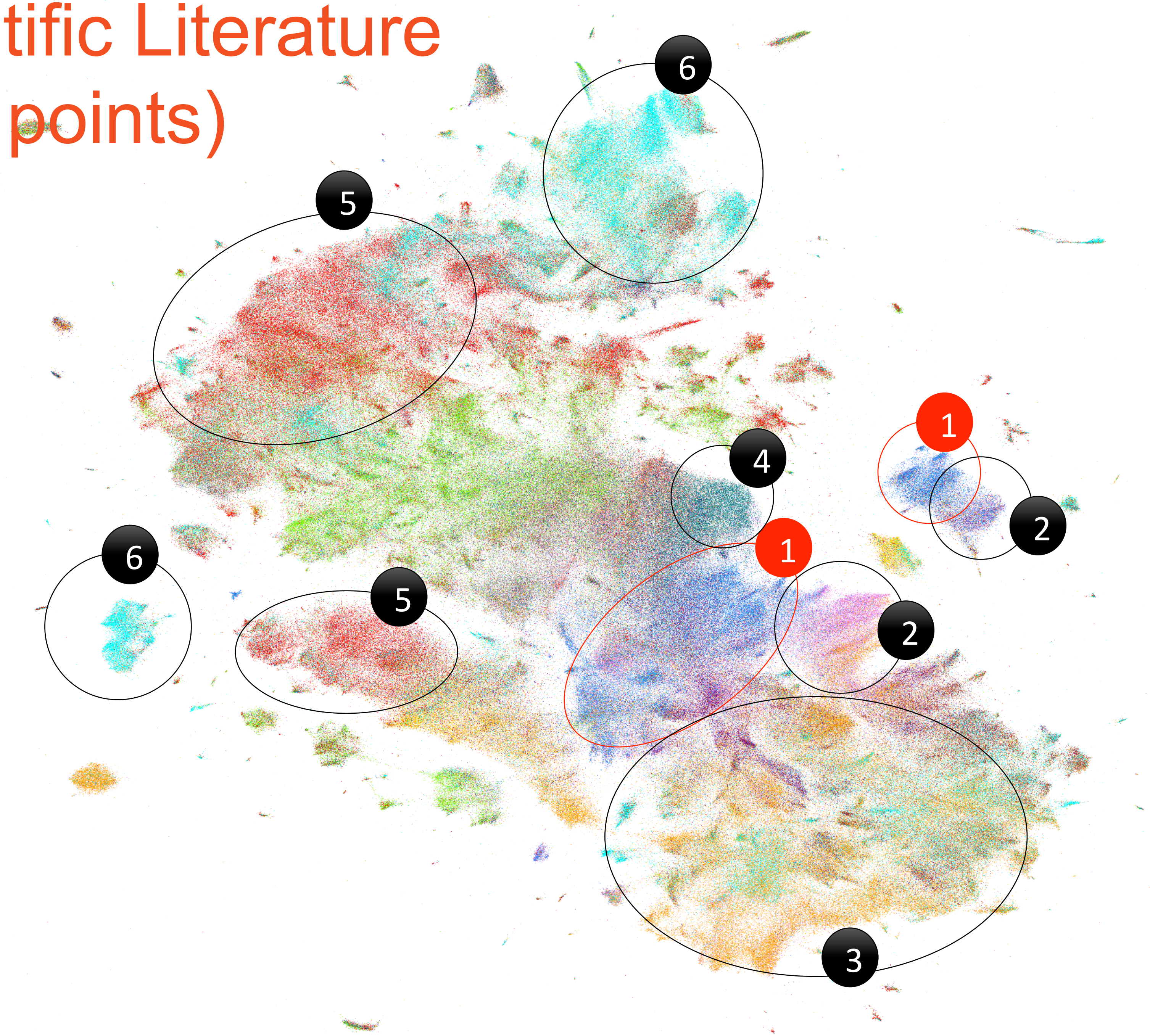
2 Mathematics

3 Physics

4 Economics

5 Biology

Scientific Literature (10M points)



1 Computer Science

2 Mathematics

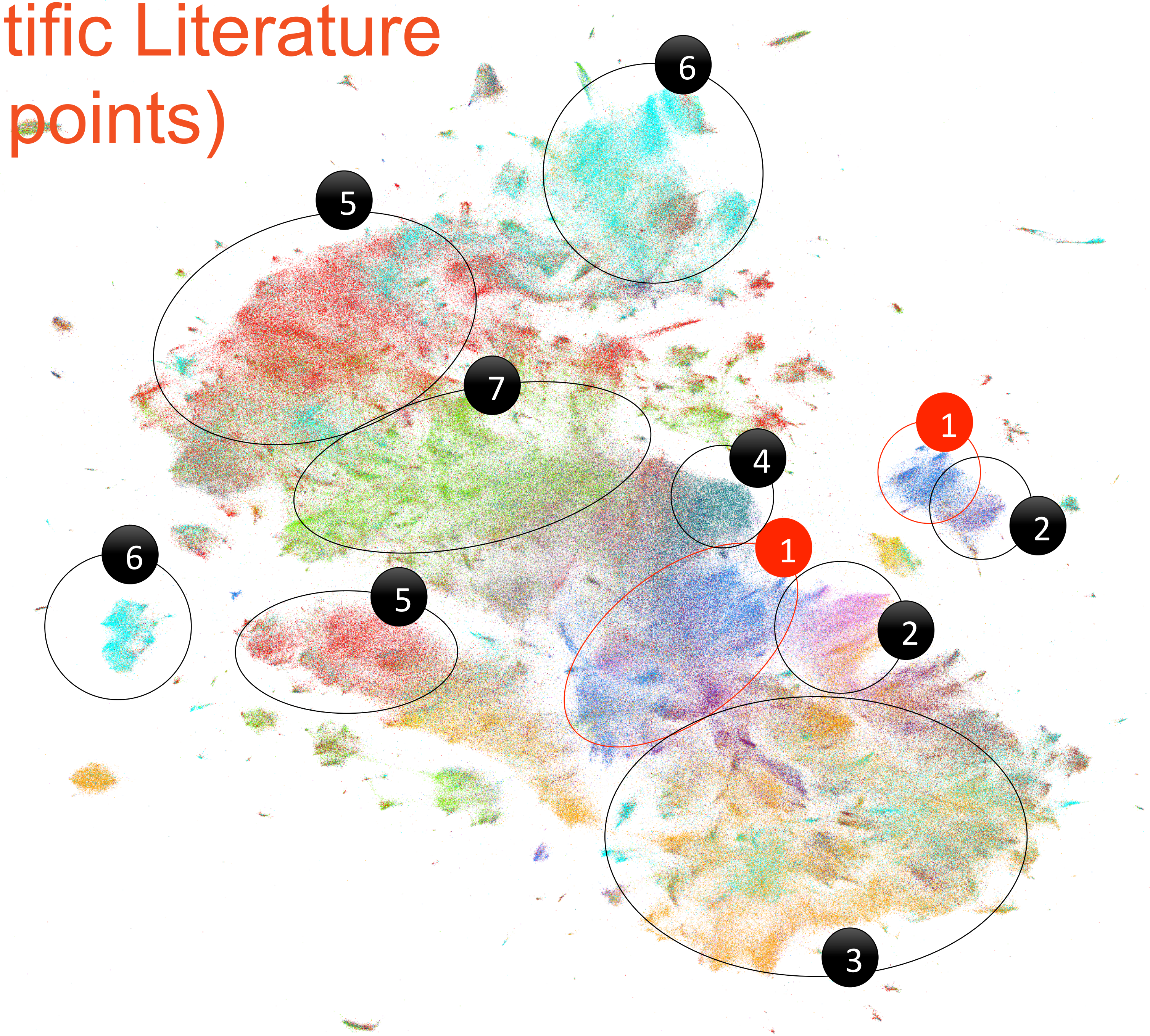
3 Physics

4 Economics

5 Biology

6 Chemistry

Scientific Literature (10M points)



1 Computer Science

2 Mathematics

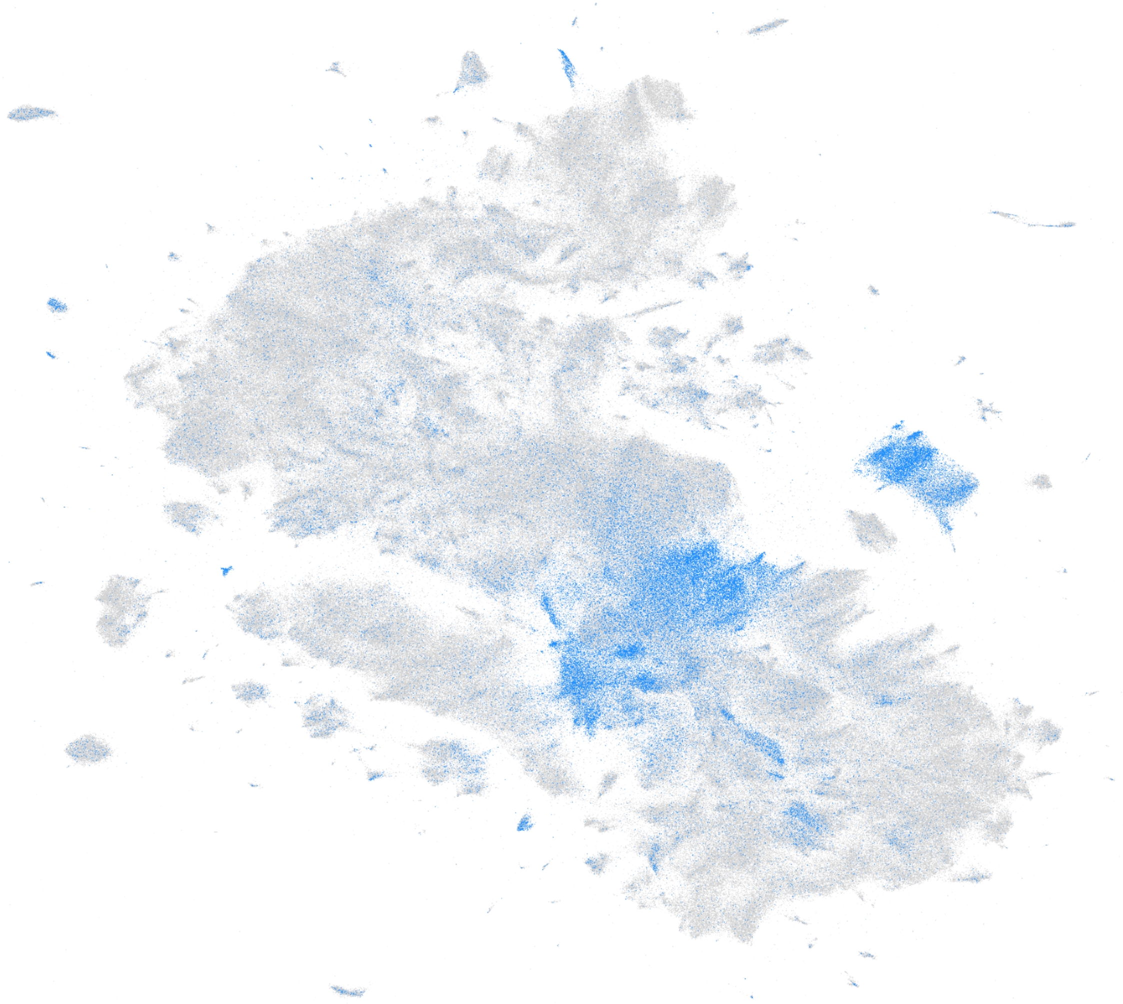
3 Physics

4 Economics

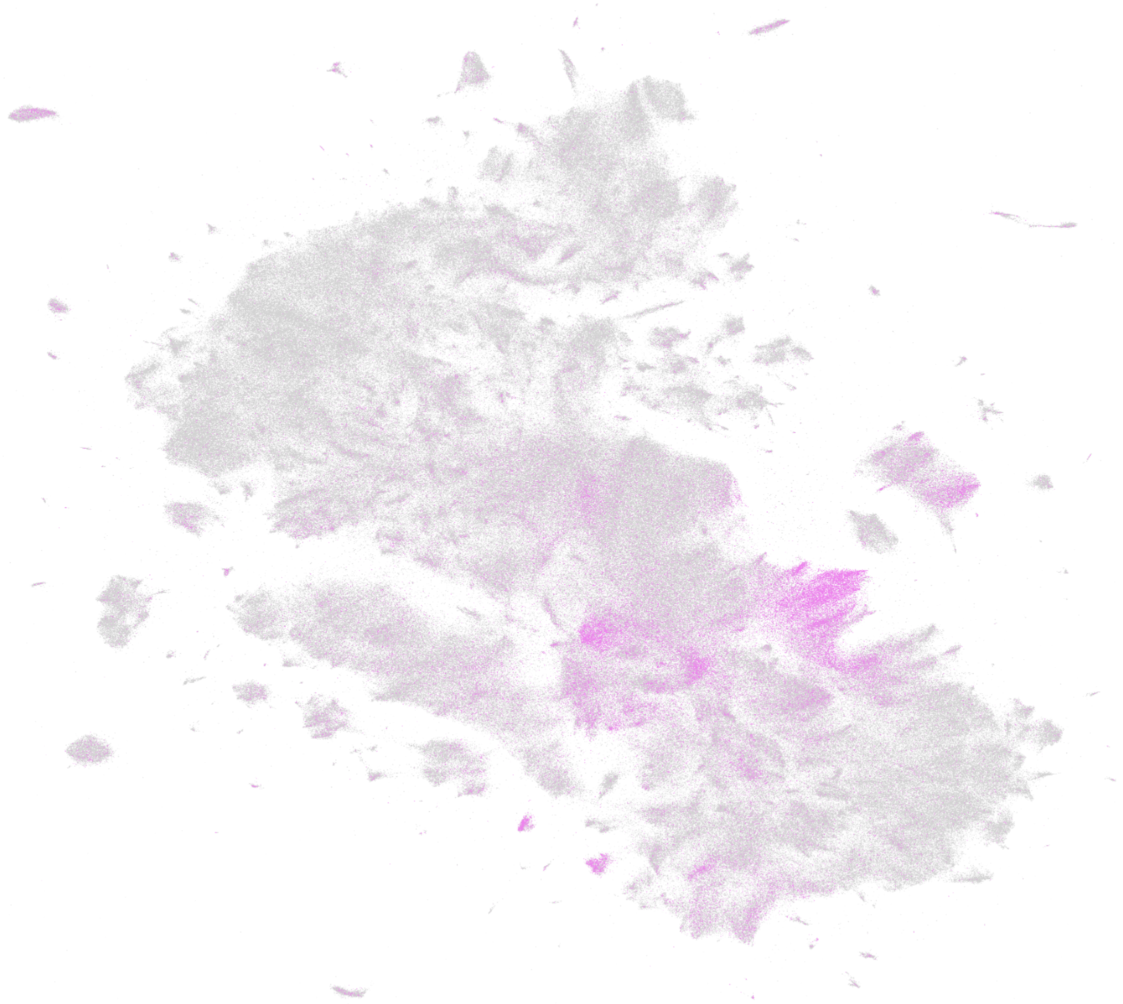
5 Biology

6 Chemistry

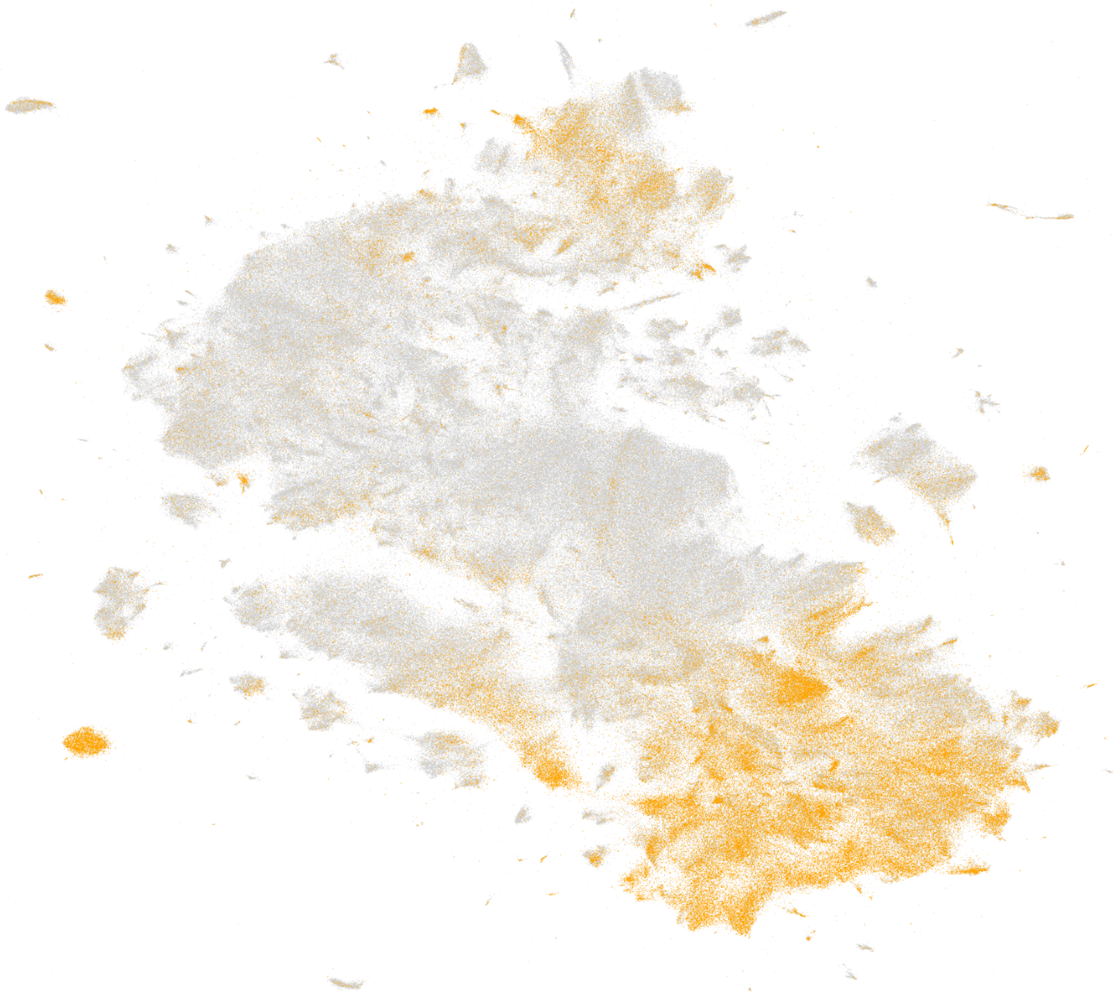
7 Medicine



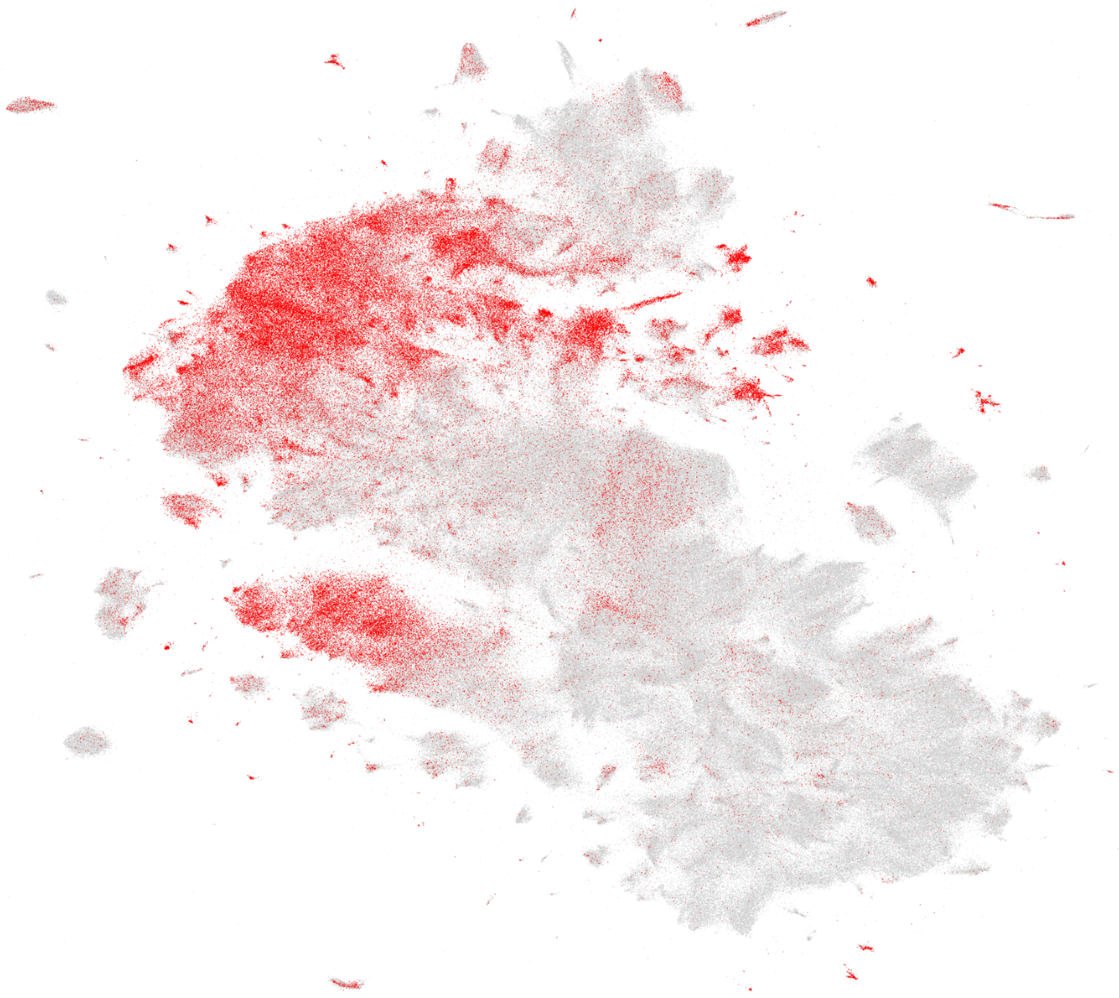
Computer Science



Mathematics



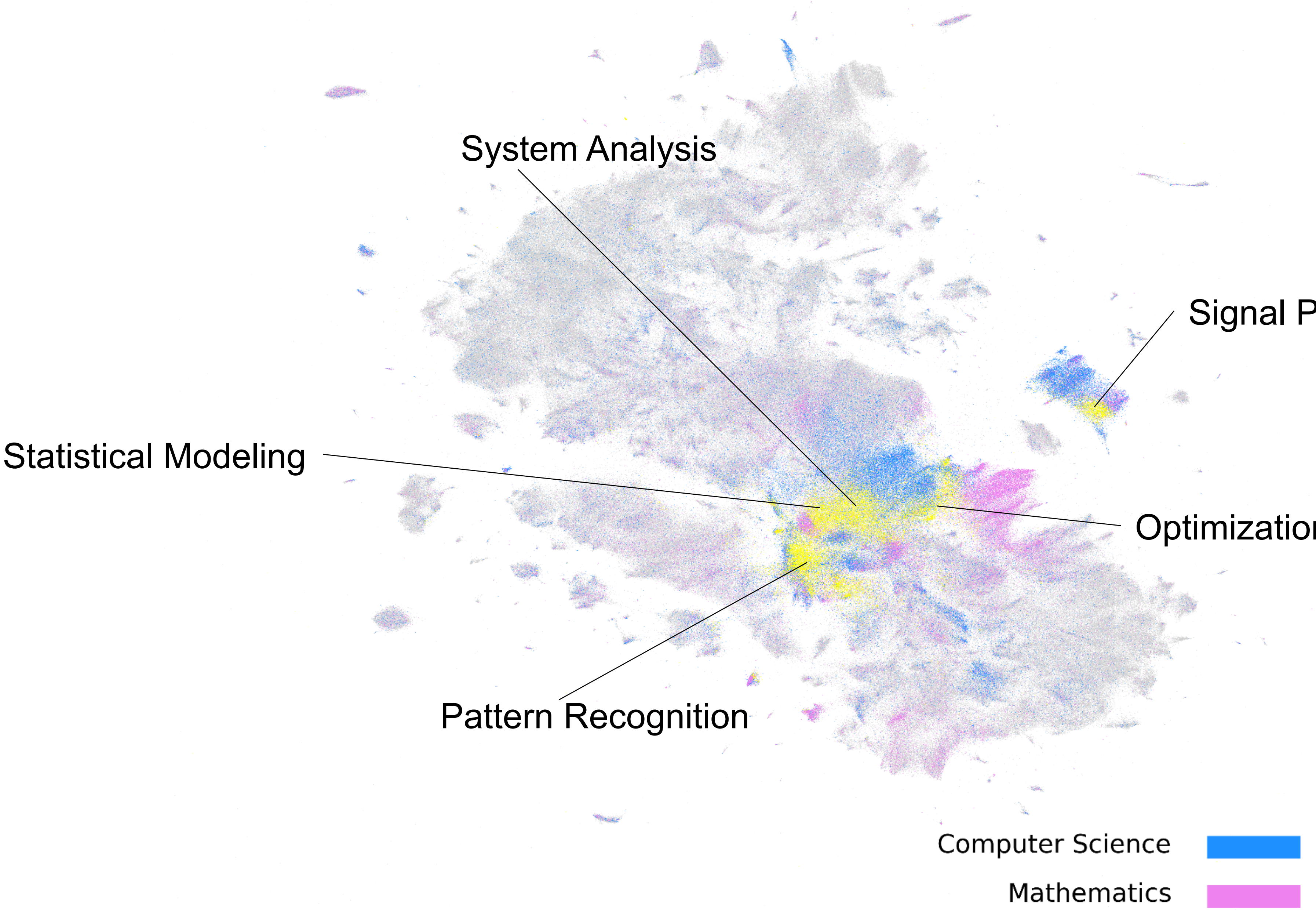
Physics



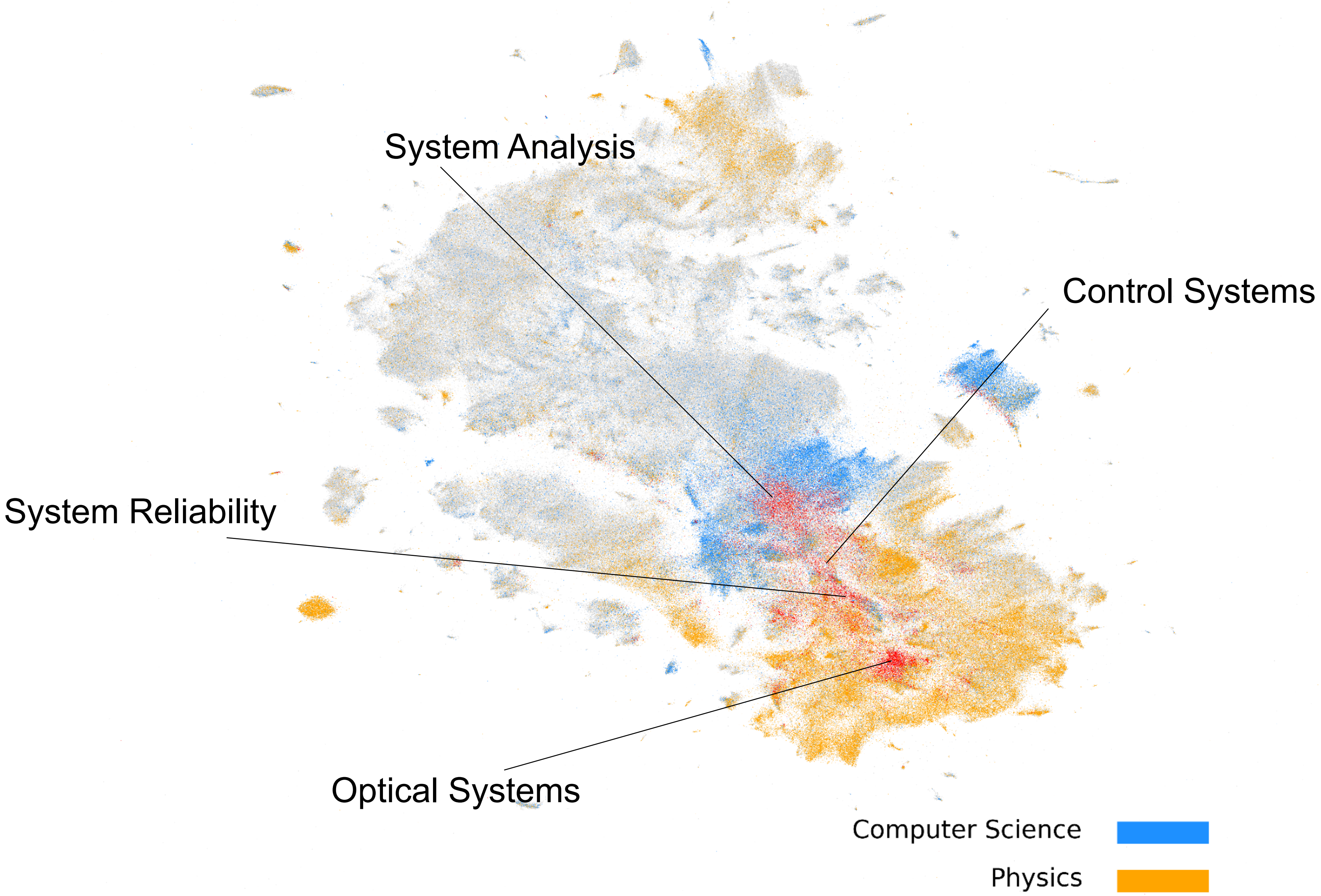
Biology

Computer Science vs. Mathematics

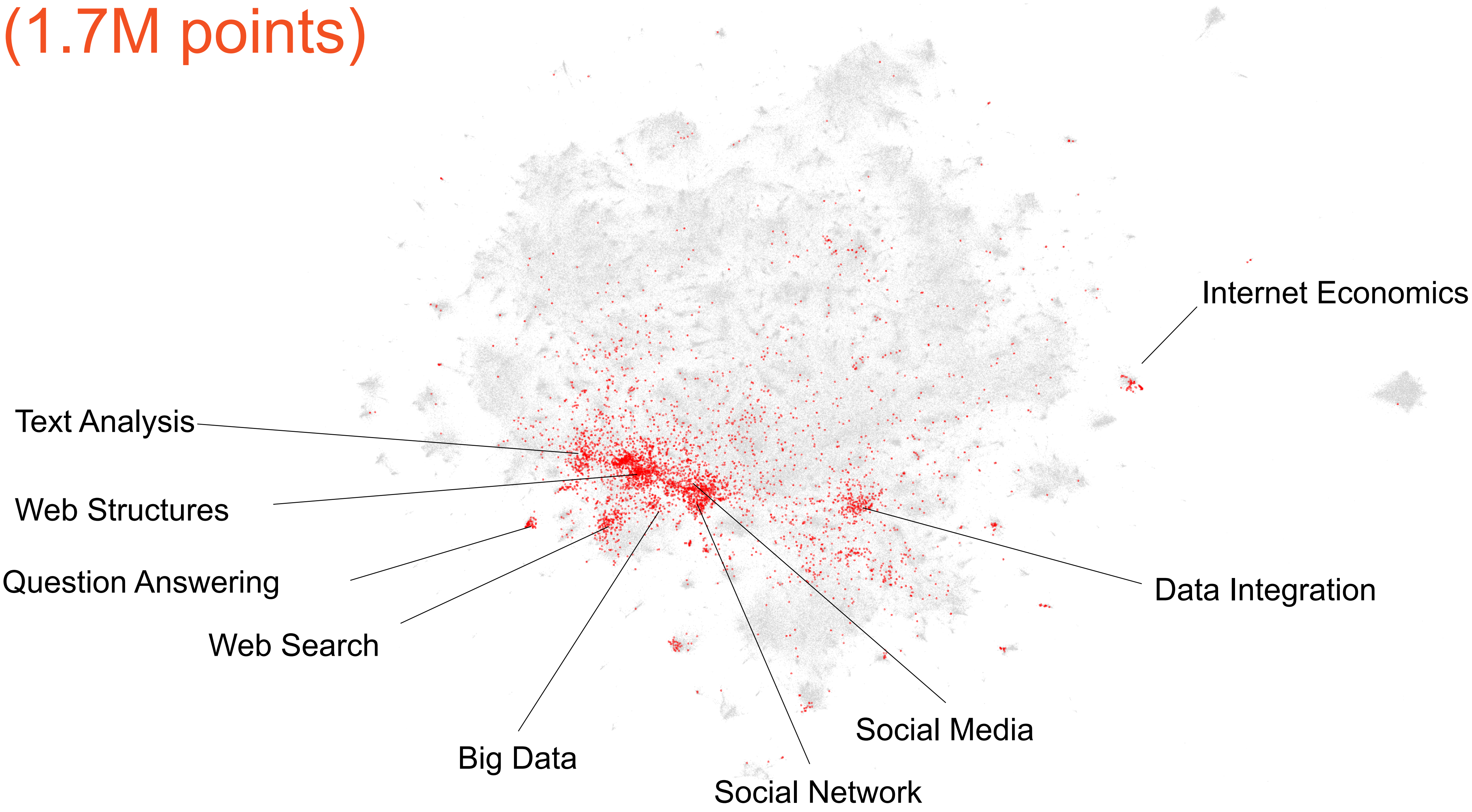
Microsoft



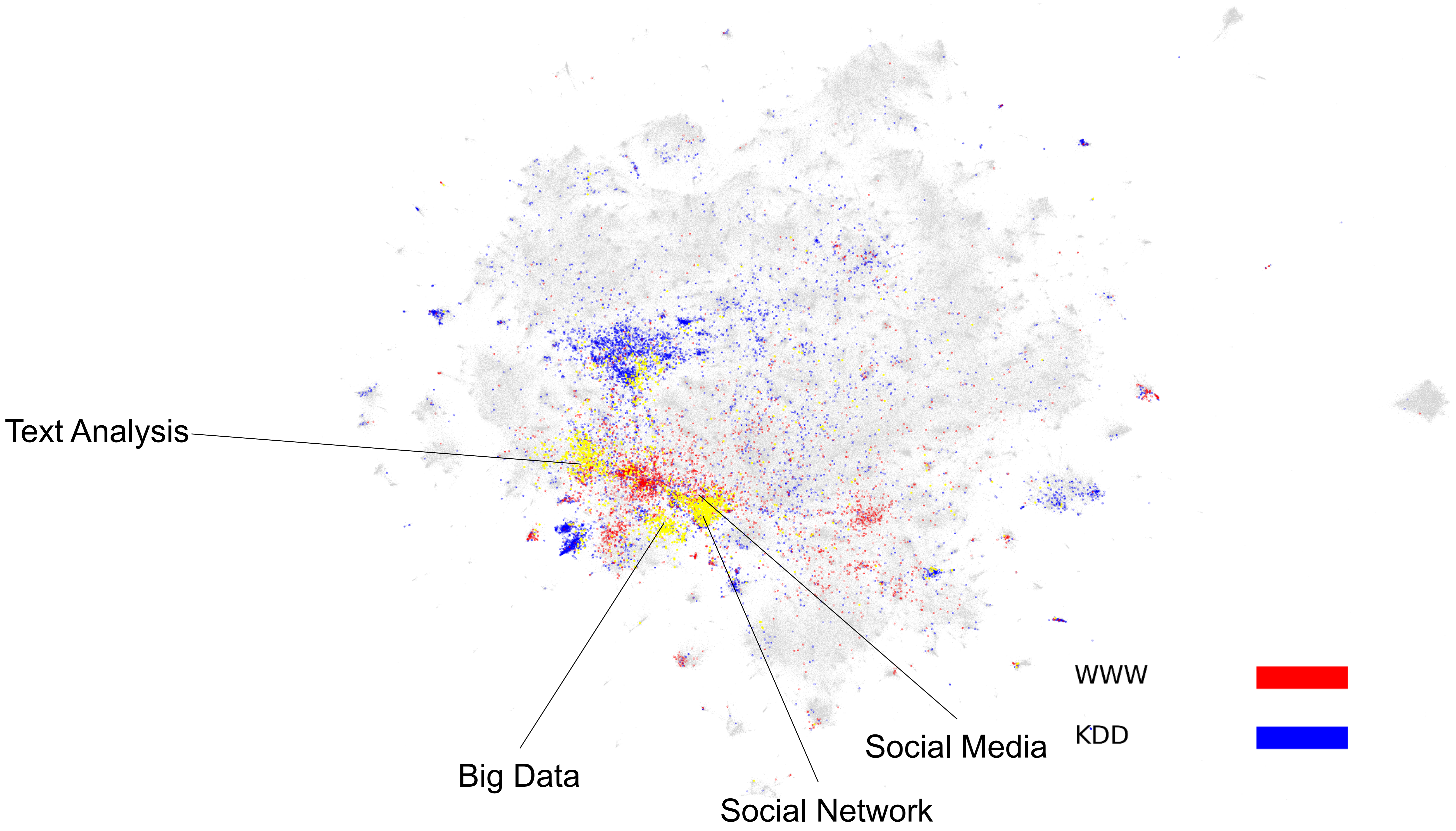
Computer Science vs. Physics



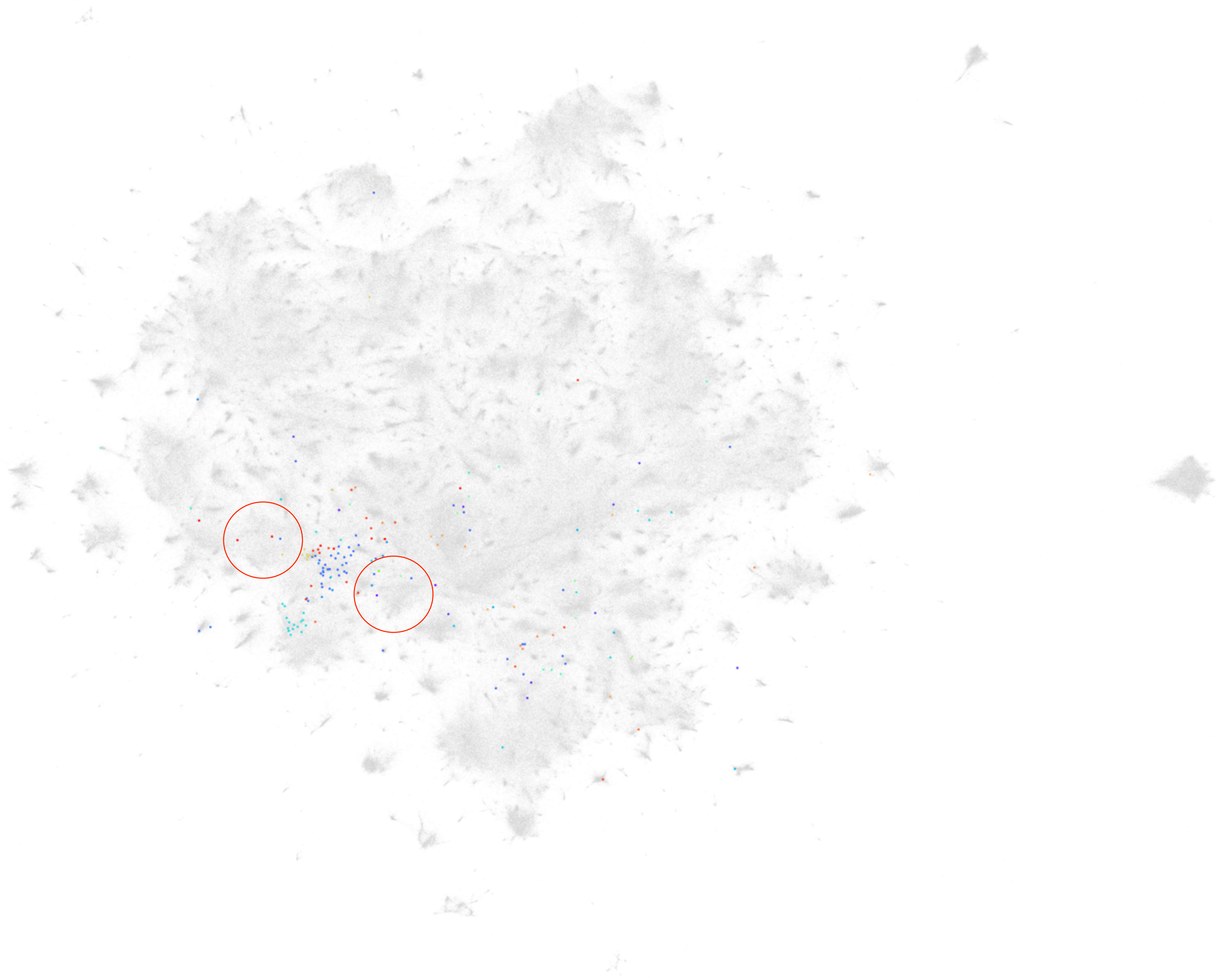
WWW in Computer Science (1.7M points)



WWW v.s. KDD



Topic Evolution: WWW2001



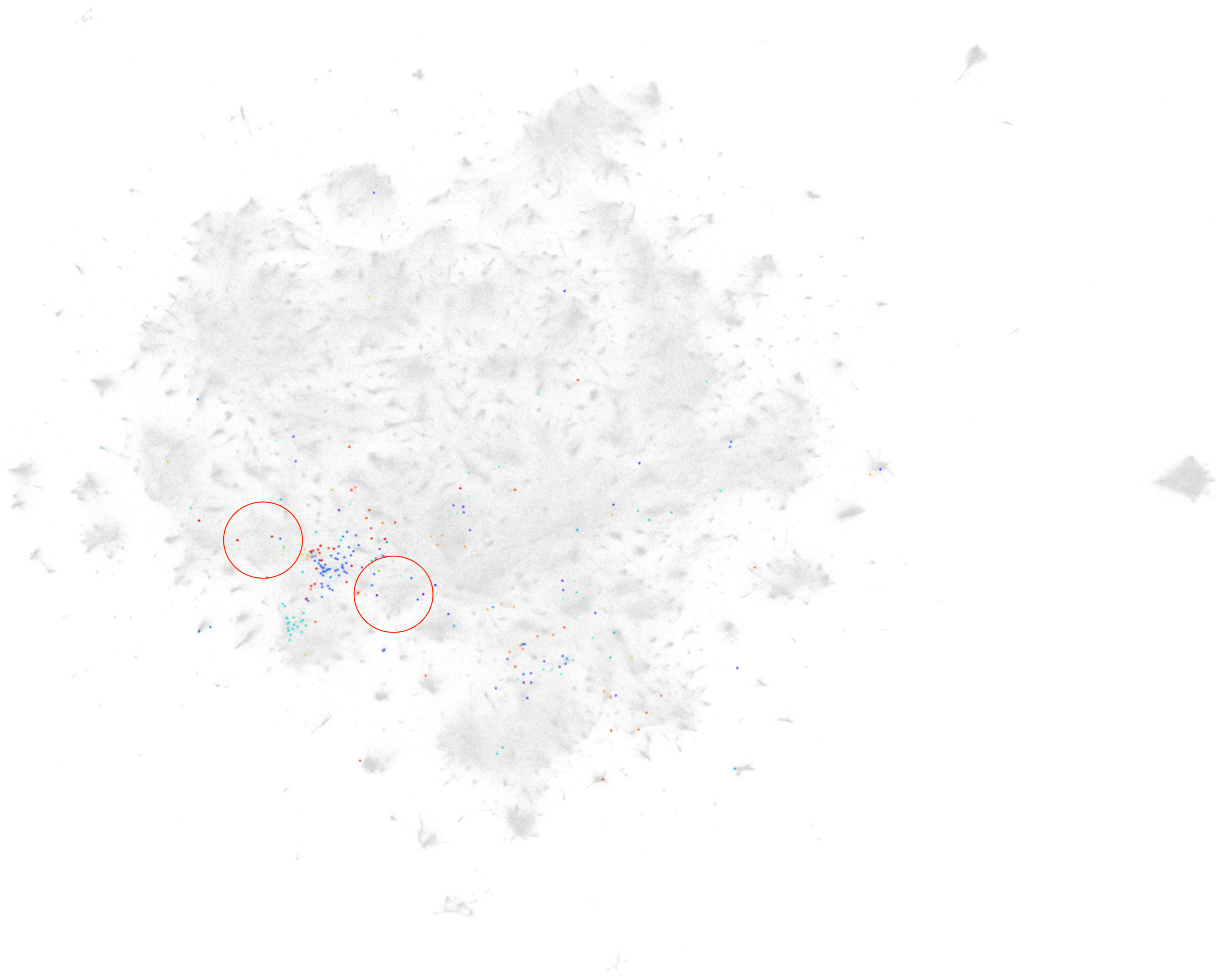
🟪 Social Network

🟩 Social Media

🟦 Web Structure

🟥 Text Analysis

Topic Evolution: WWW2002



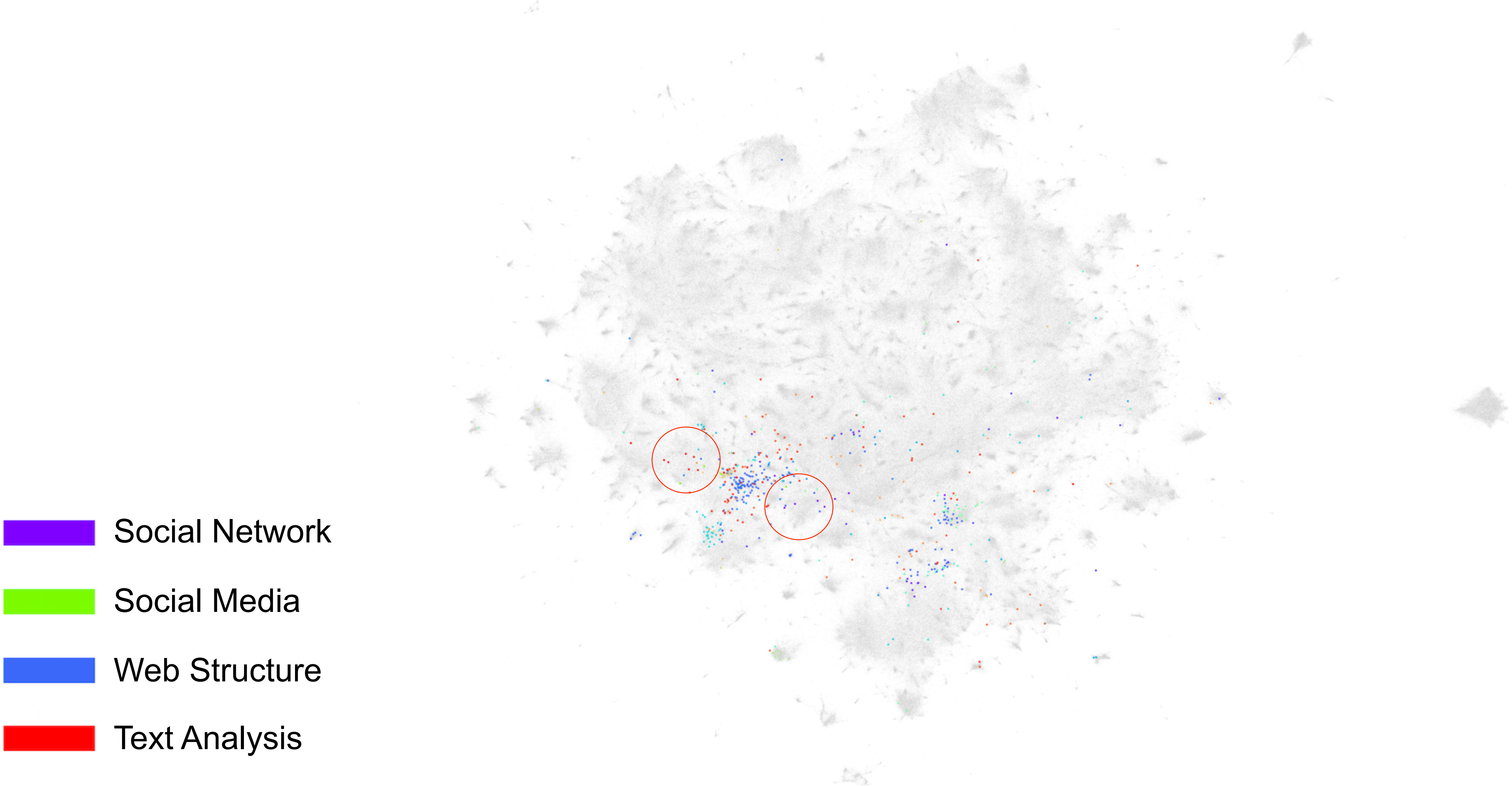
🟪 Social Network

🟩 Social Media

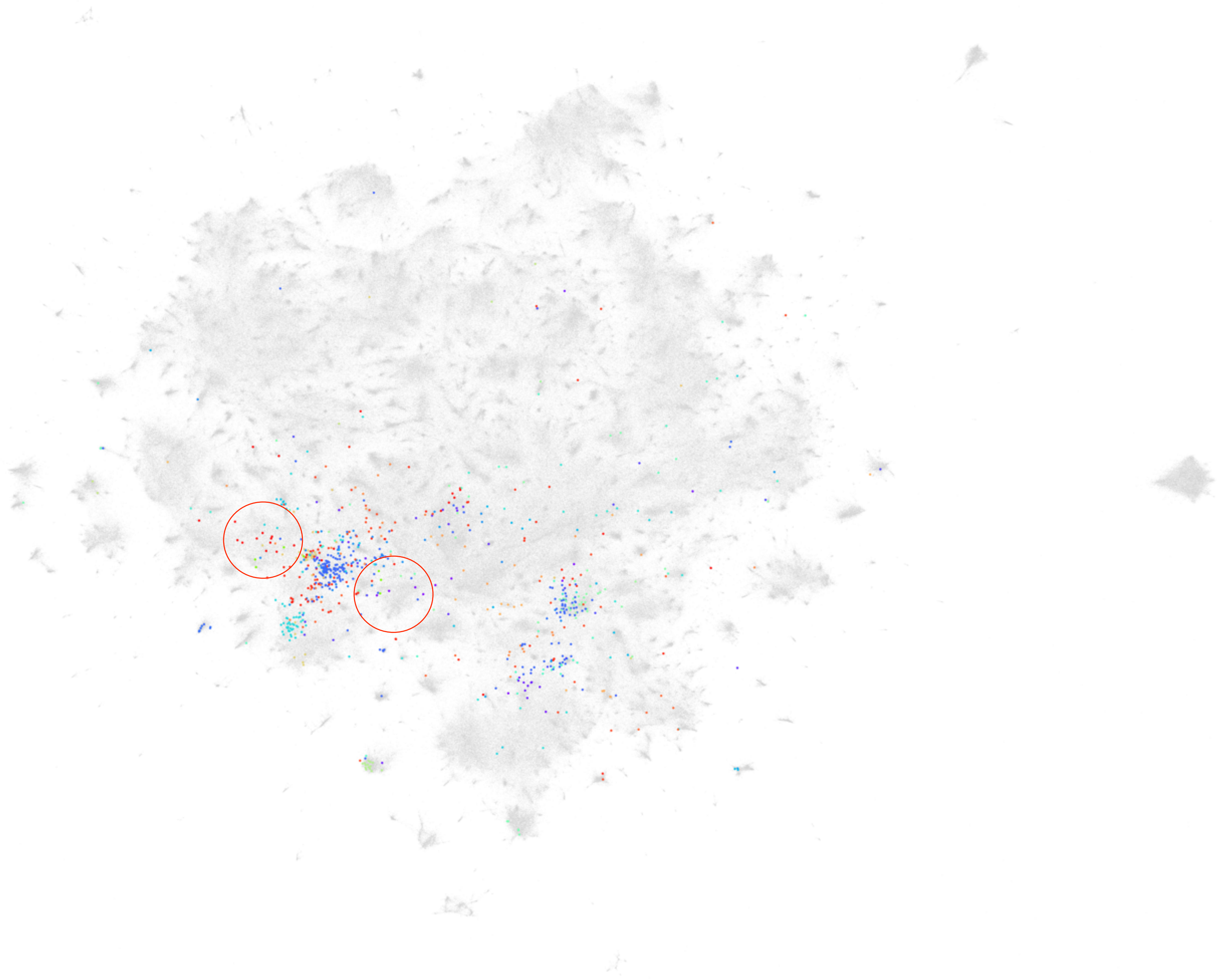
🟦 Web Structure

🟥 Text Analysis

Topic Evolution: WWW2003



Topic Evolution: WWW2004



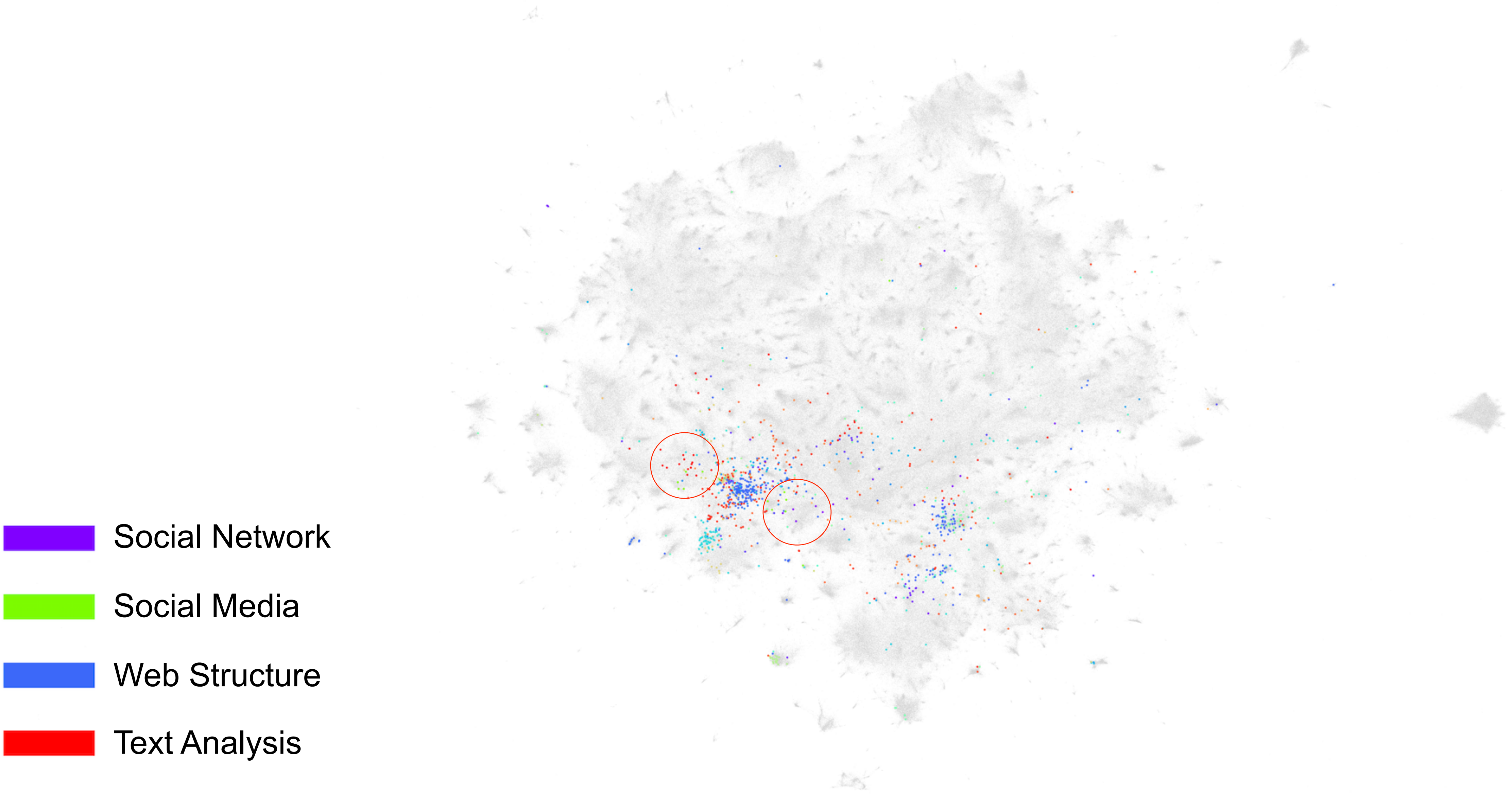
🟪 Social Network

🟩 Social Media

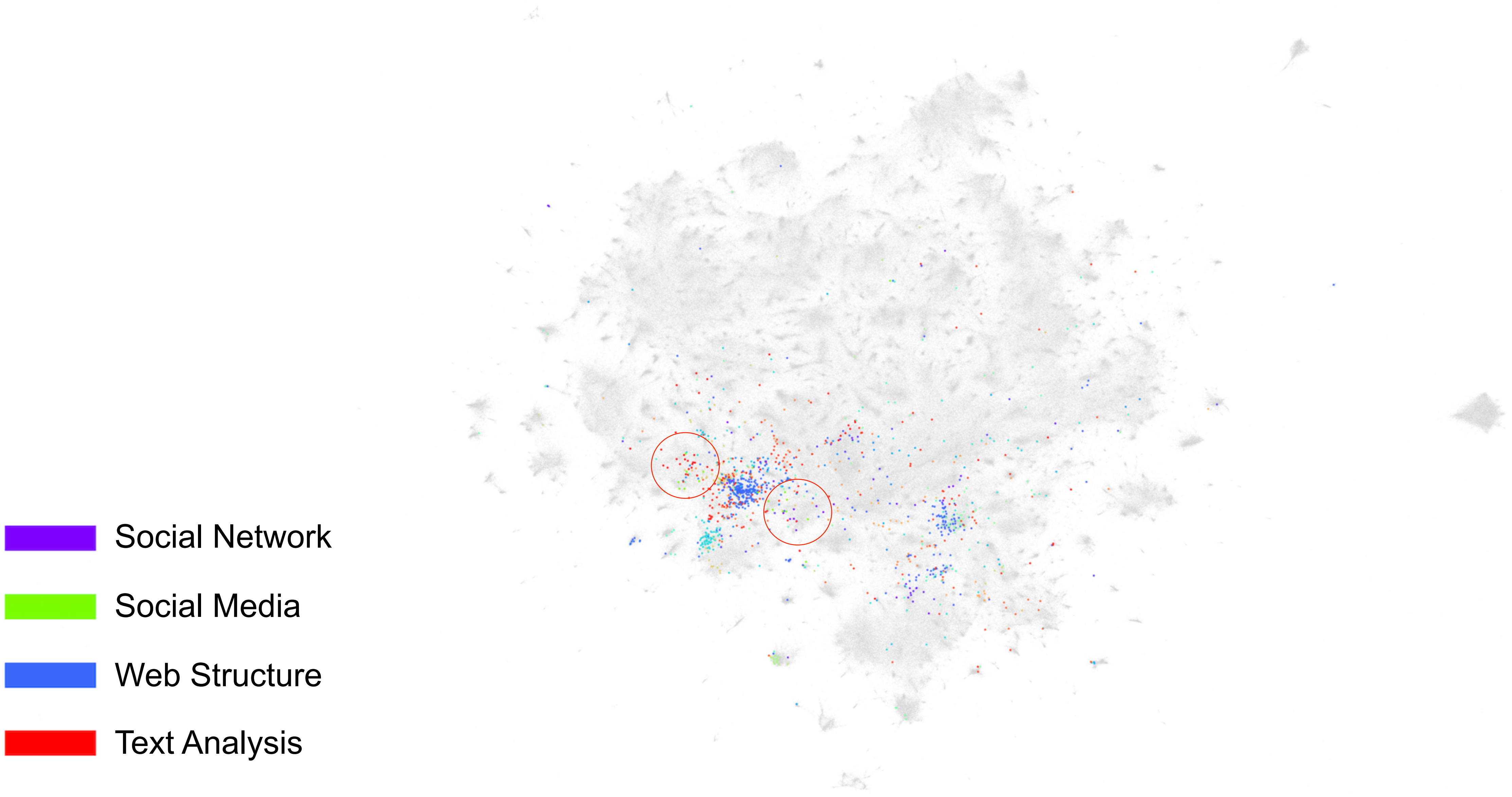
🟦 Web Structure

🟥 Text Analysis

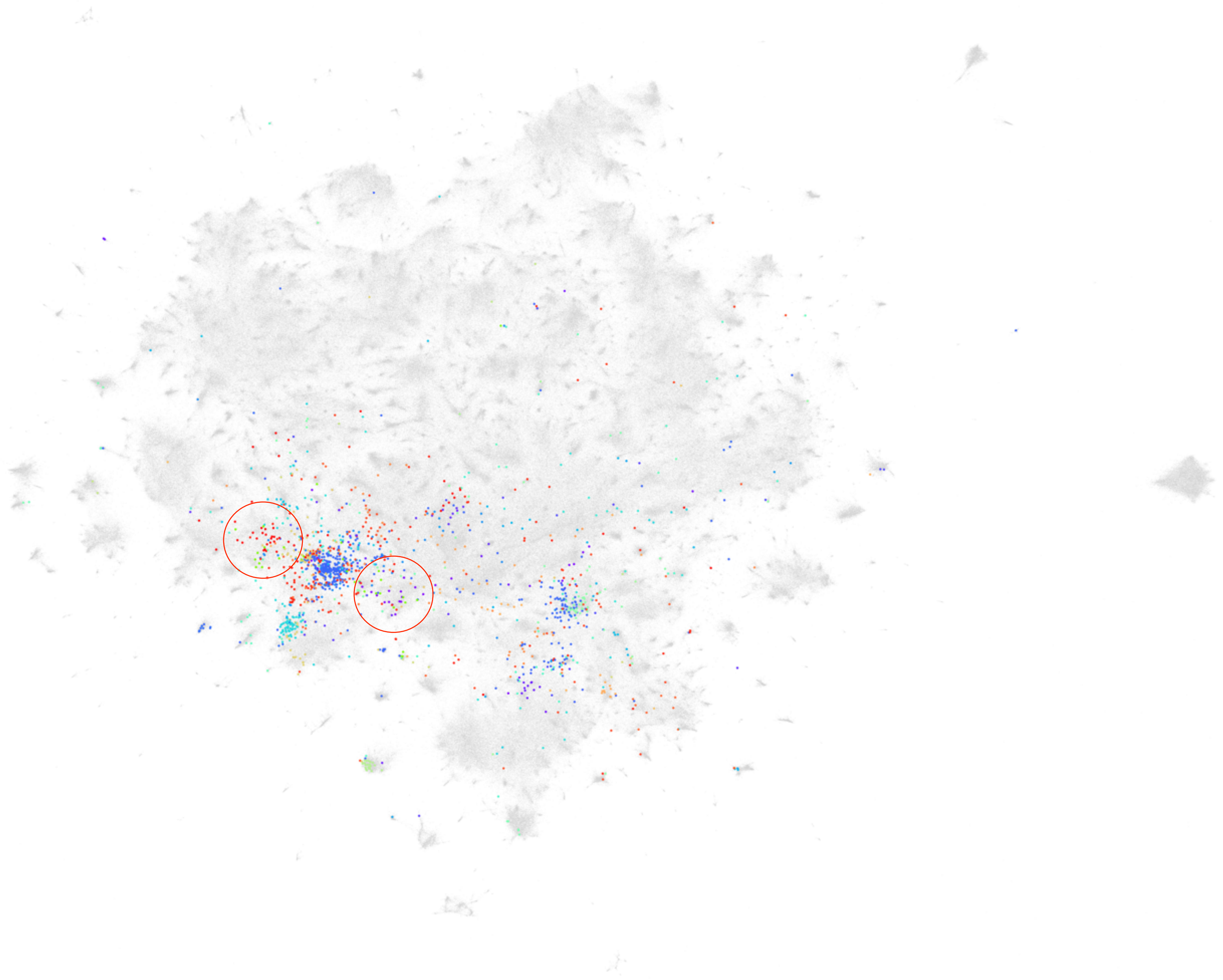
Topic Evolution: WWW2005



Topic Evolution: WWW2006



Topic Evolution: WWW2007



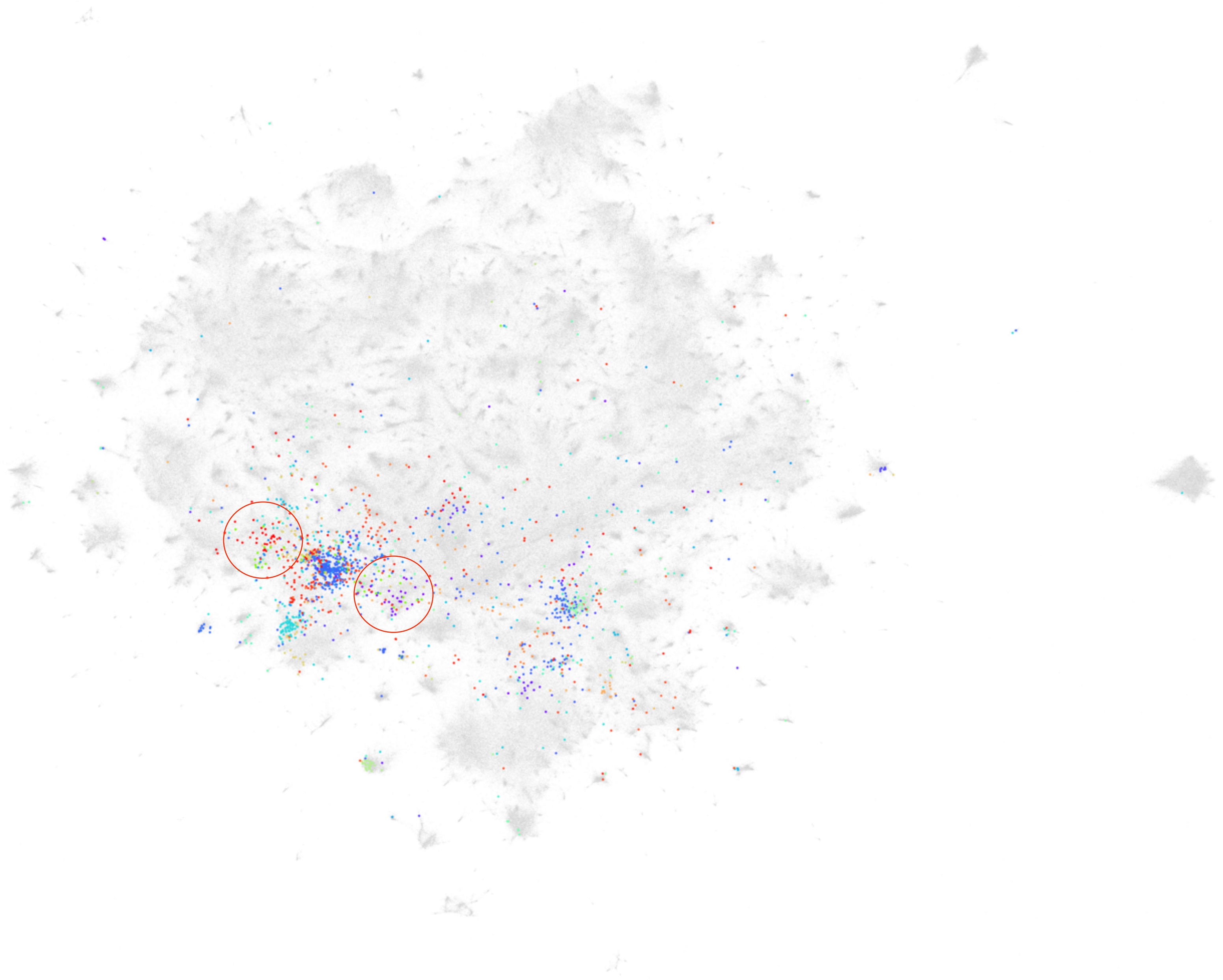
🟪 Social Network

🟩 Social Media

🟦 Web Structure

🟥 Text Analysis

Topic Evolution: WWW2008



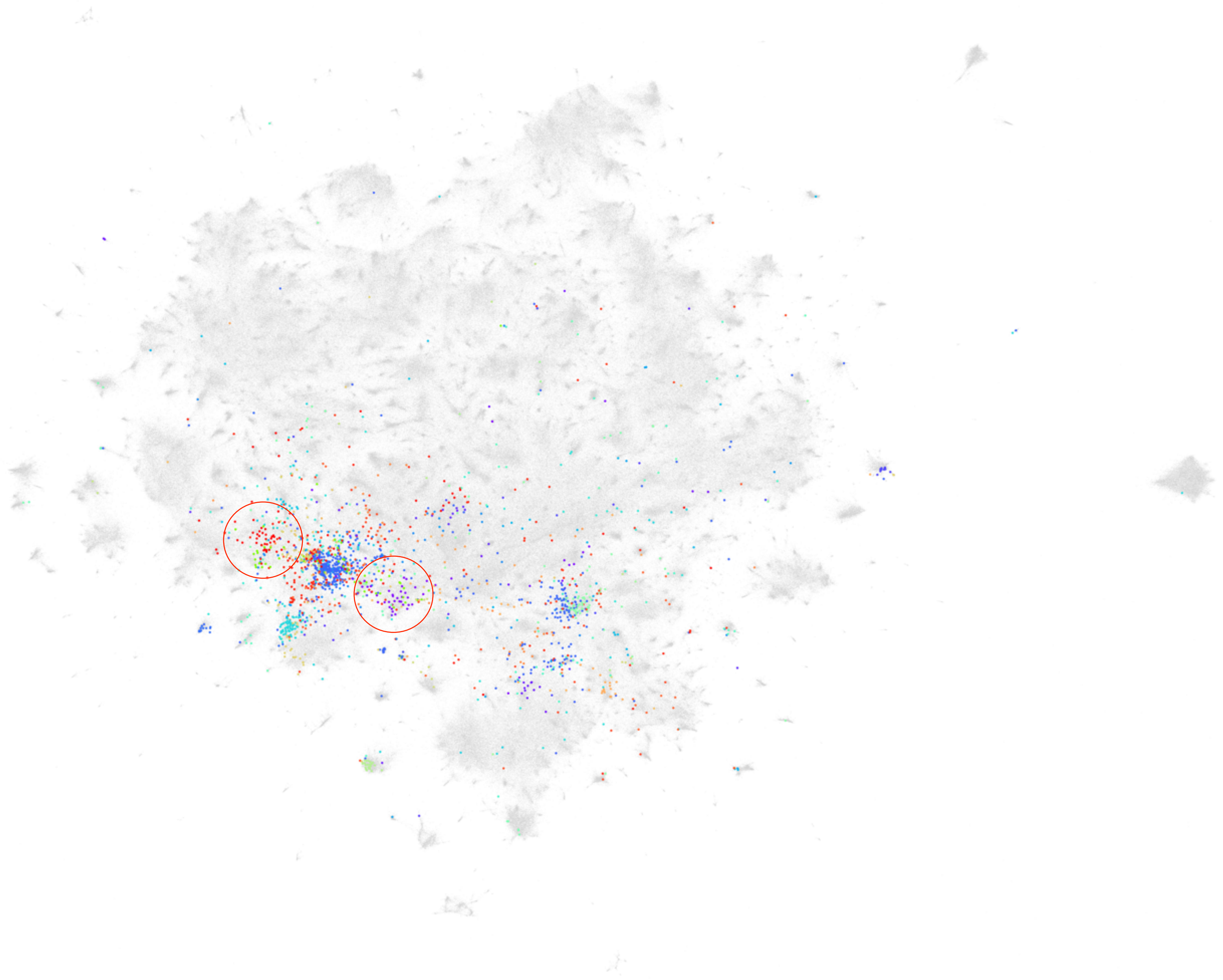
🟪 Social Network

🟩 Social Media

🟦 Web Structure

🟥 Text Analysis

Topic Evolution: WWW2009



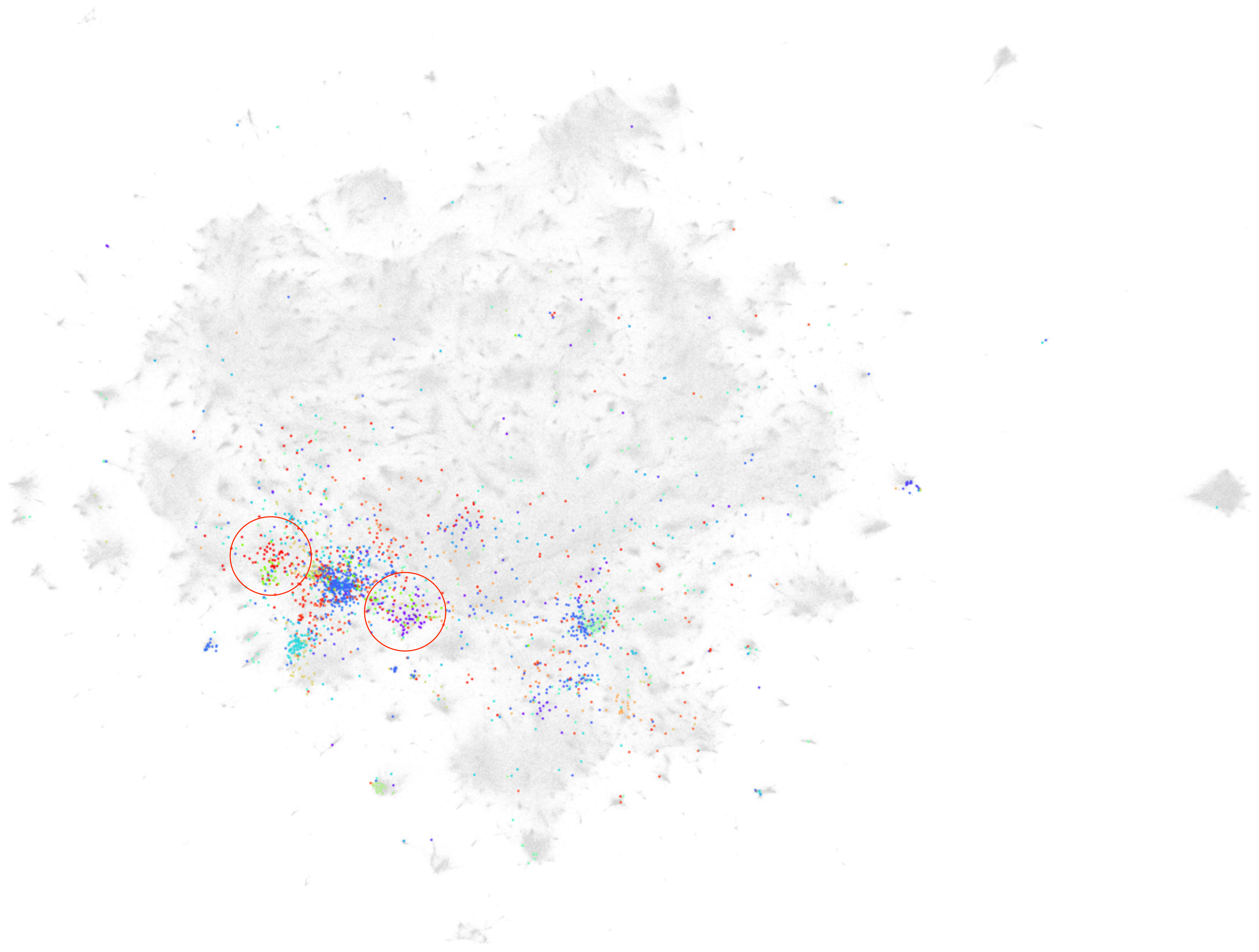
🟪 Social Network

🟩 Social Media

🟦 Web Structure

🟥 Text Analysis

Topic Evolution: WWW2010



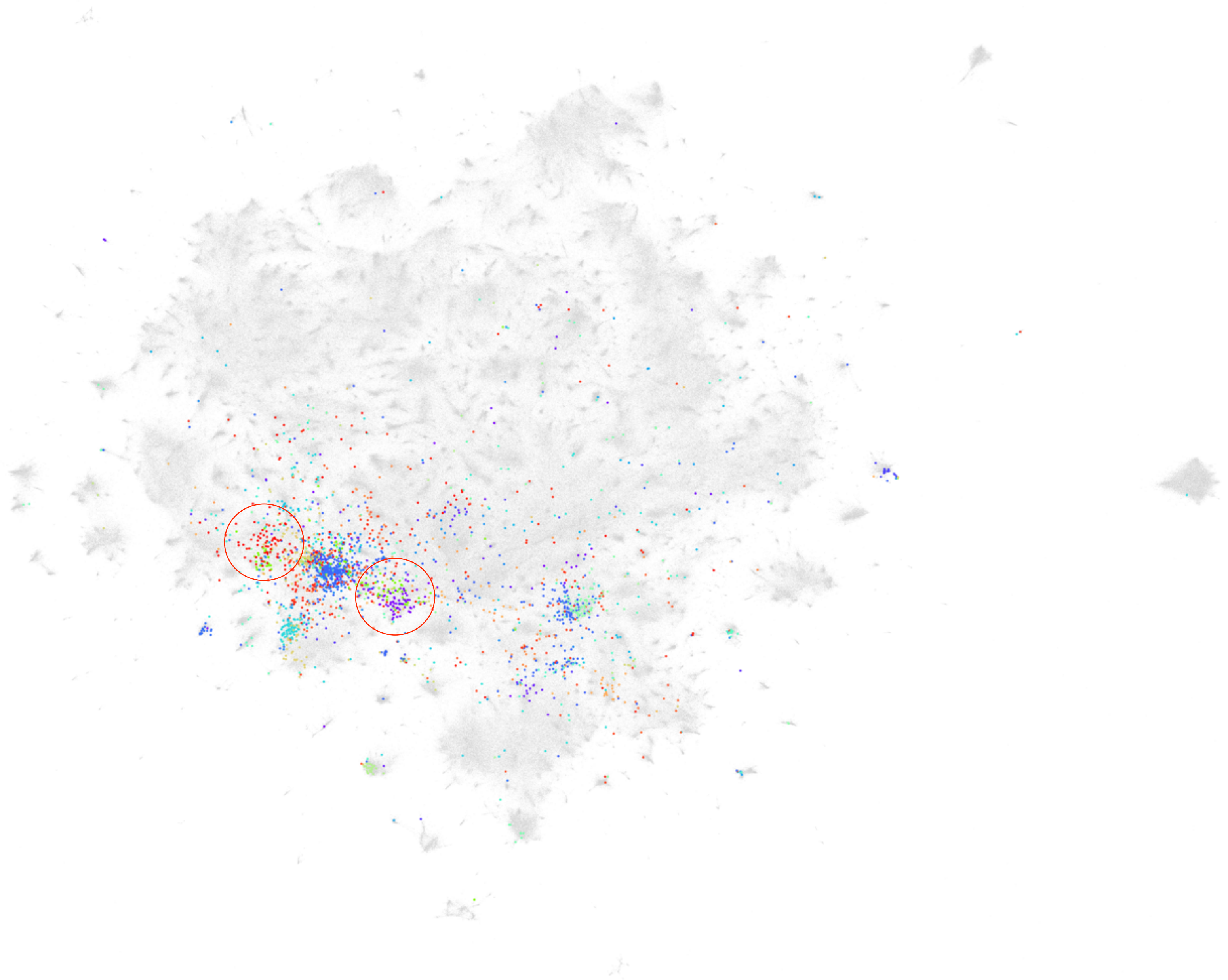
🟪 Social Network

🟩 Social Media

🟦 Web Structure

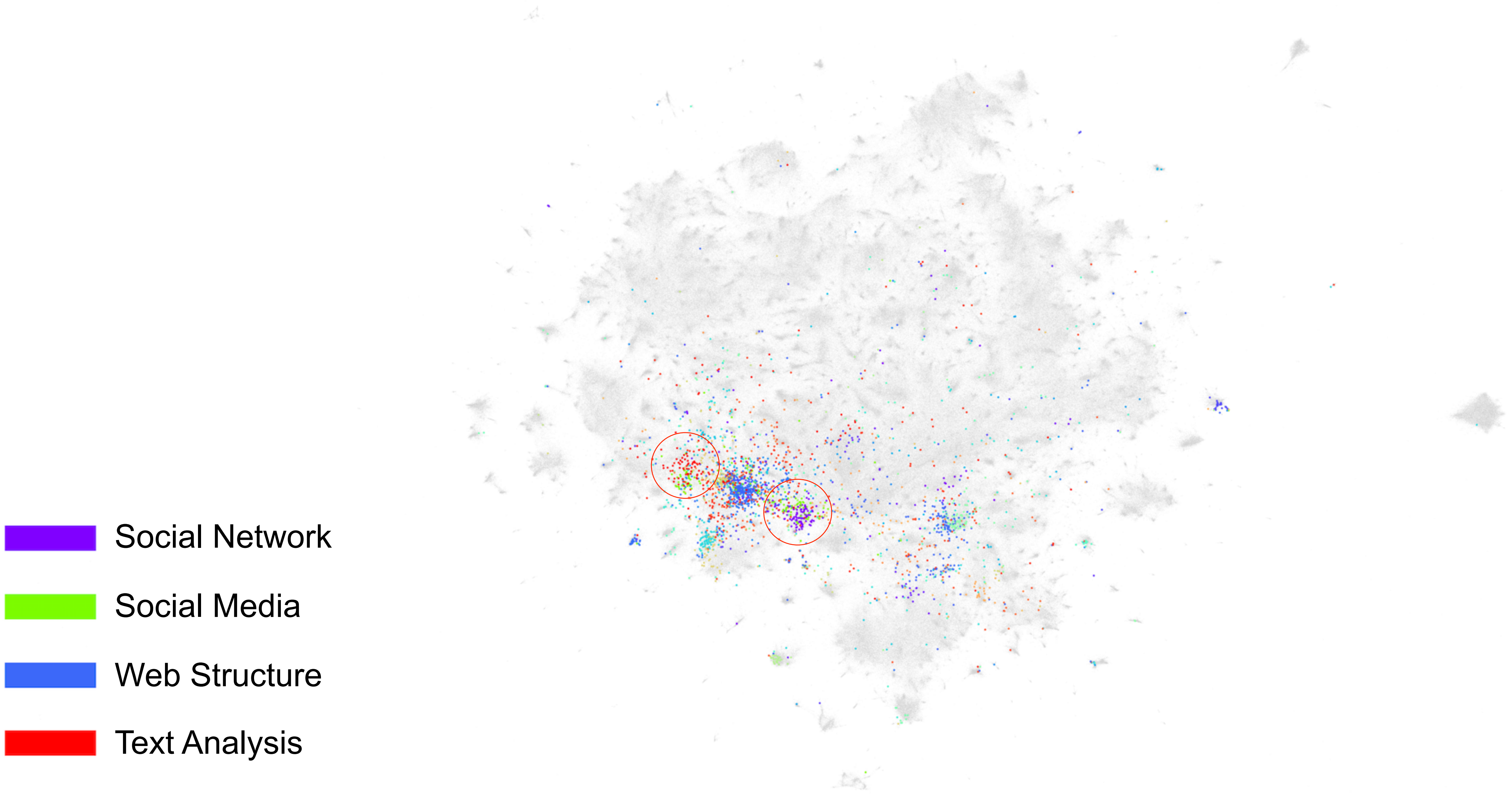
🟥 Text Analysis

Topic Evolution: WWW2011

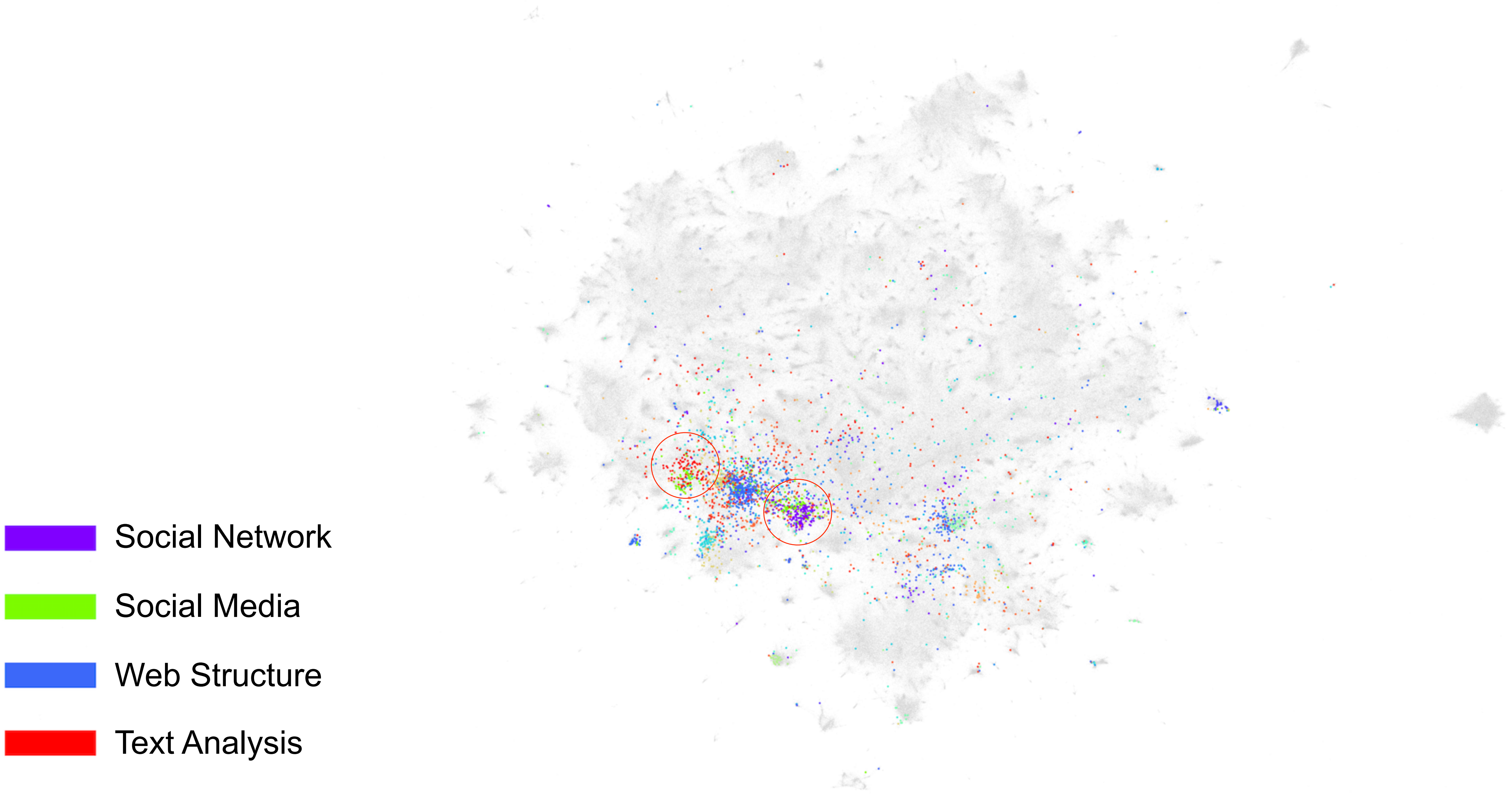


- 🟪 Social Network
- 🟩 Social Media
- 🟦 Web Structure
- 🟥 Text Analysis

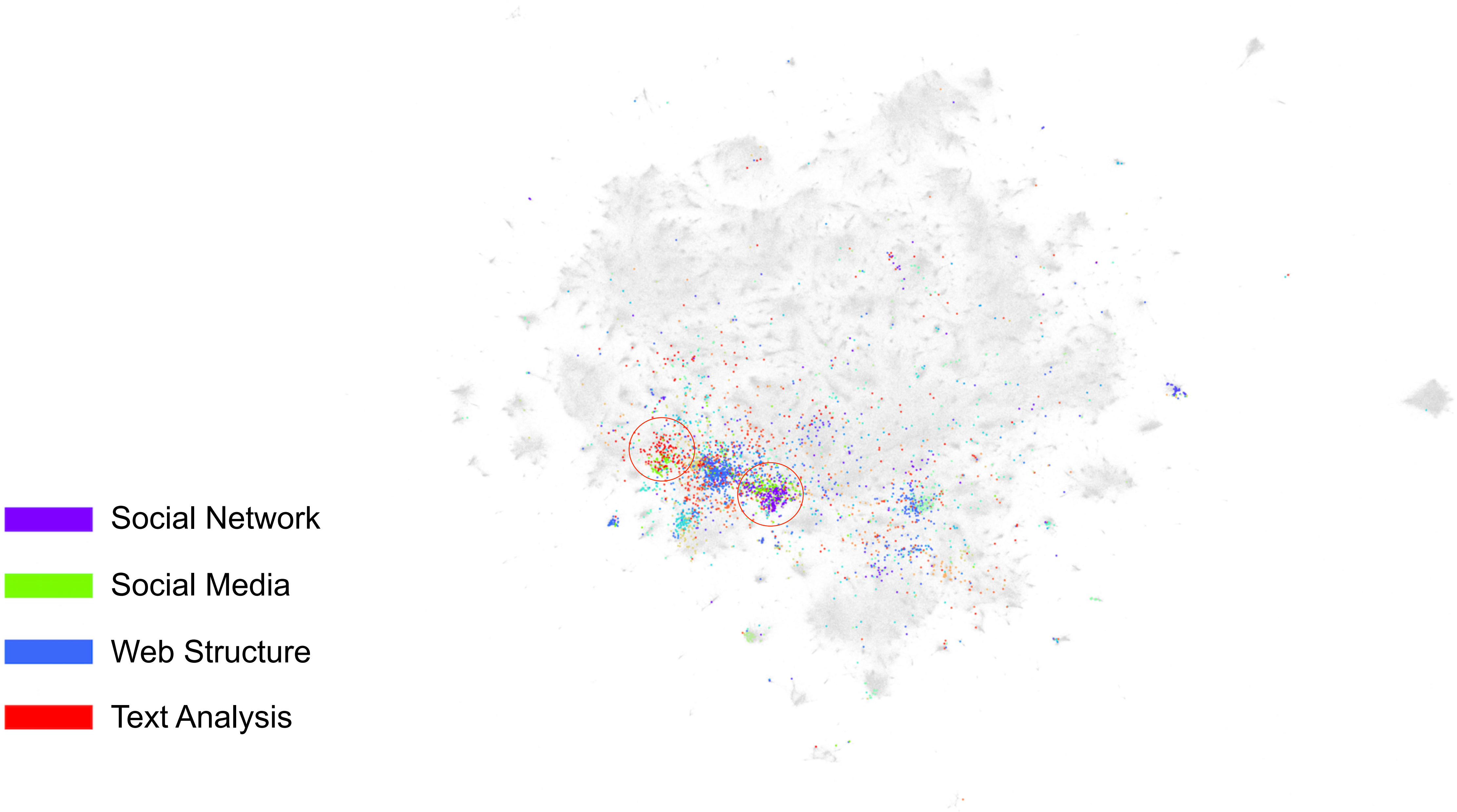
Topic Evolution: WWW2012



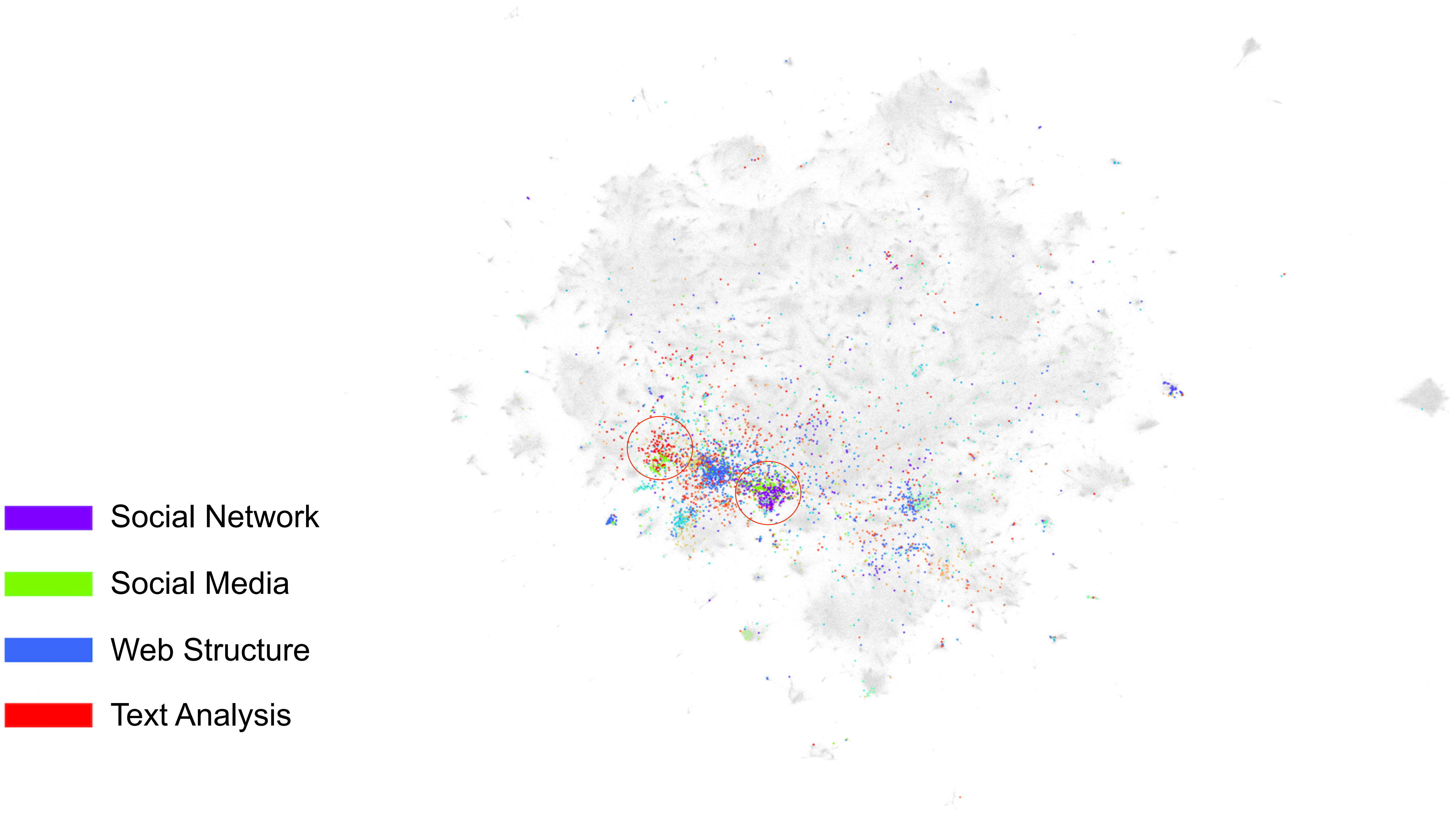
Topic Evolution: WWW2013



Topic Evolution: WWW2014

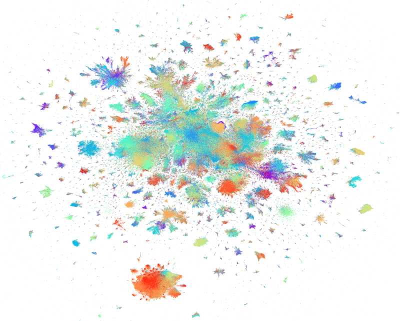


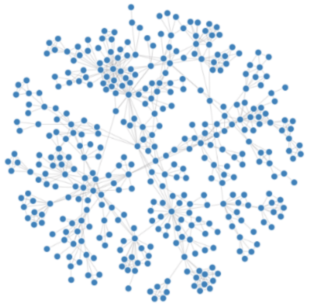
Topic Evolution: WWW2015

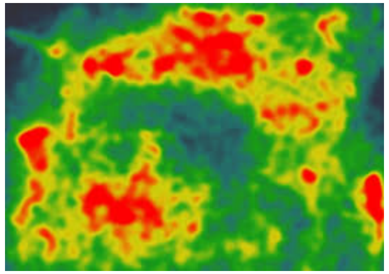


Challenging to Visualize the Big Data

- *Intuitive* ways for data understanding and exploration
- Classical visualization techniques
  - Scatter plots, network diagrams, heatmaps,...
  - Requires 2D/3D layouts of data
- Real-world data are often Big
  - E.g., images, text, speech and networks
  - *Large-scale* (> millions) and *high-dimensional* (> hundreds)


Scatter Plots

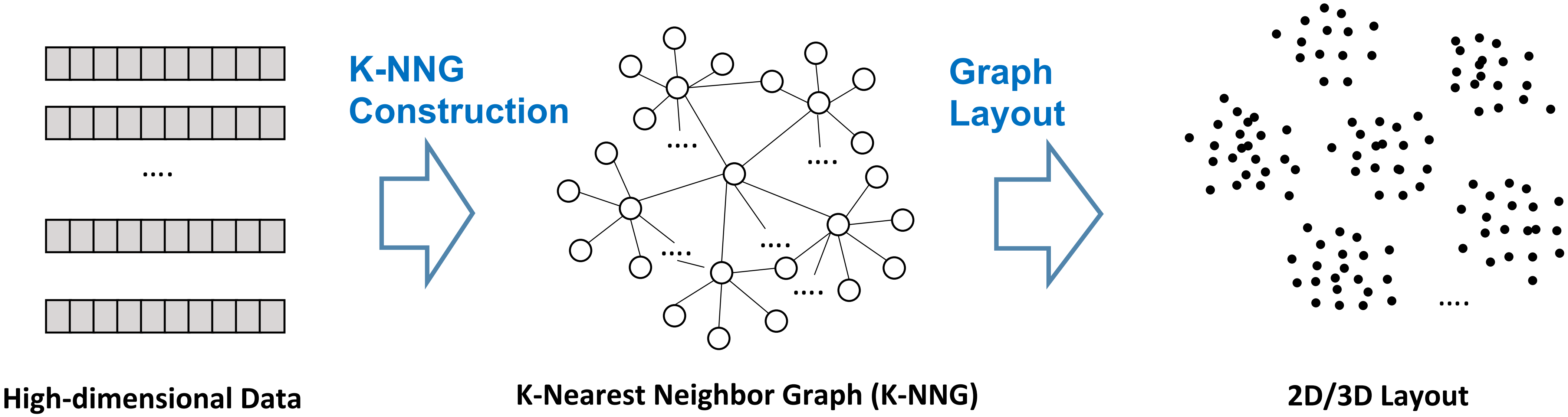

Network Diagrams


Heatmaps

Related Work

- Linear methods: e.g., PCA, MDS
  - High-dimensional data are usually on a ***nonlinear*** manifold
- Nonlinear methods: e.g., IsoMap, Laplacian Eigenmap.
  - Only preserve ***local*** structures of data
- Nonlinear method: t-SNE (Maaten and Hinton, 2008)
  - Current state-of-the-art
  - Preserve both ***local*** and ***global*** structures
  - But difficult to scale up

Typical Pipeline of Data Visualization



High-dimensional Data

K-NNG Construction

K-Nearest Neighbor Graph (K-NNG)

Graph Layout

2D/3D Layout

- Limitations of t-SNE:
  - K-NNG construction: complexity grows *exponentially* to the data dimension
  - Graph layout: complexity is *O(NlogN),* where N is the number of data points
  - Very *sensitive* parameters

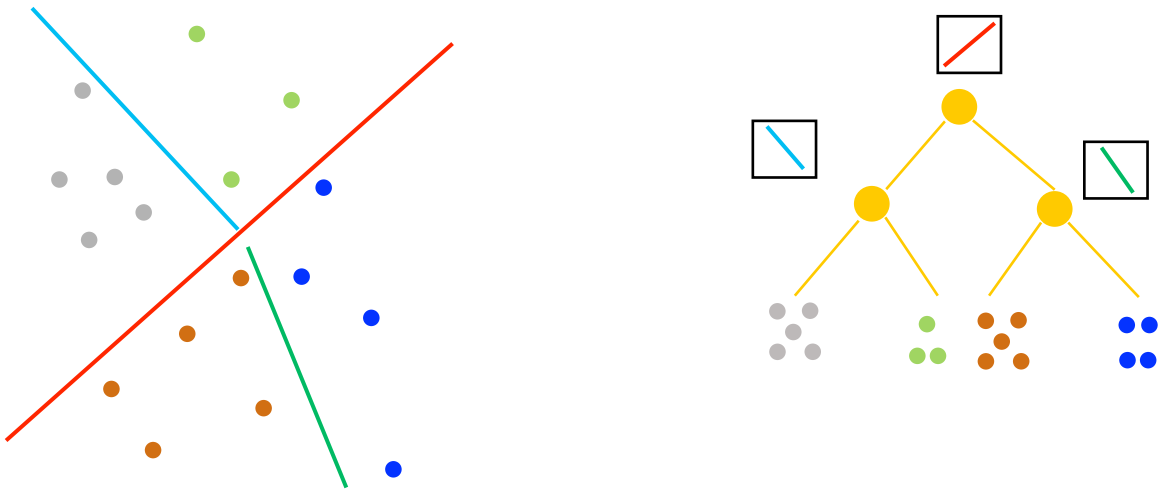
Our Approach: **LargeVis**

- An efficient approach for approximate K-NNG construction
  - ***Thirty*** times faster than t-SNE on 3 million data points
  - Better time-accuracy tradeoff

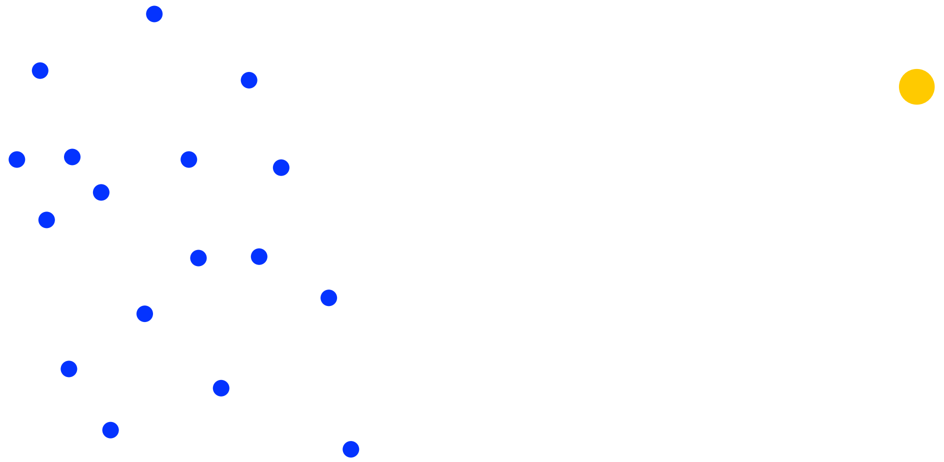
- An efficient probabilistic model for graph layout
  - O(NlogN) -> O(N)
  - ***Seven*** times faster than t-SNE on 3 million data points
  - More ***effective*** visualization layouts than t-SNE
  - ***Stable*** parameters across different data sets

Random Projection Trees

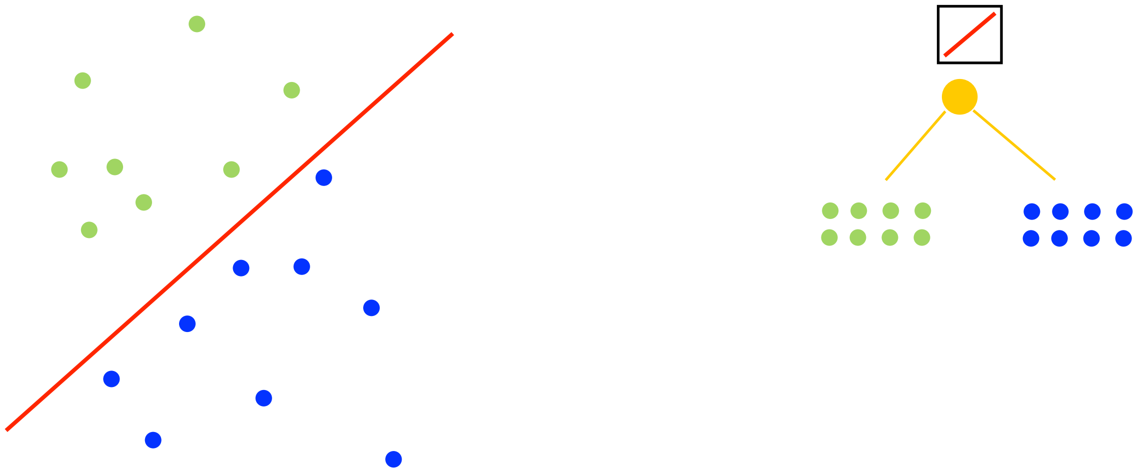
- Partition the whole space into different regions with multiple hyperplanes



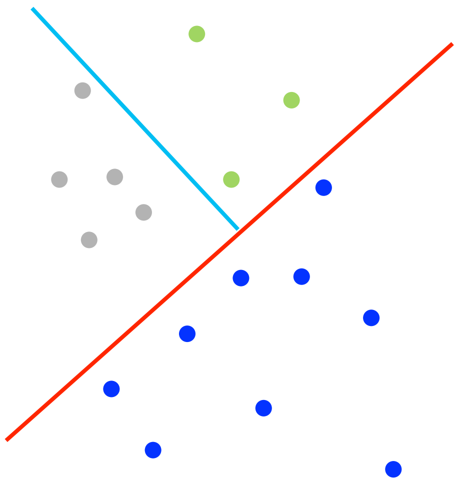
Random Projection Trees

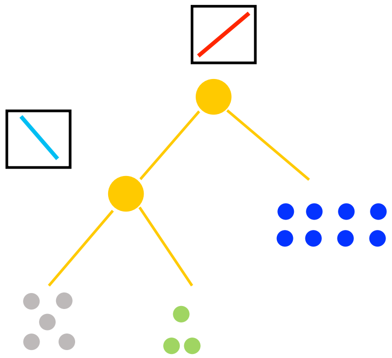


Random Projection Trees

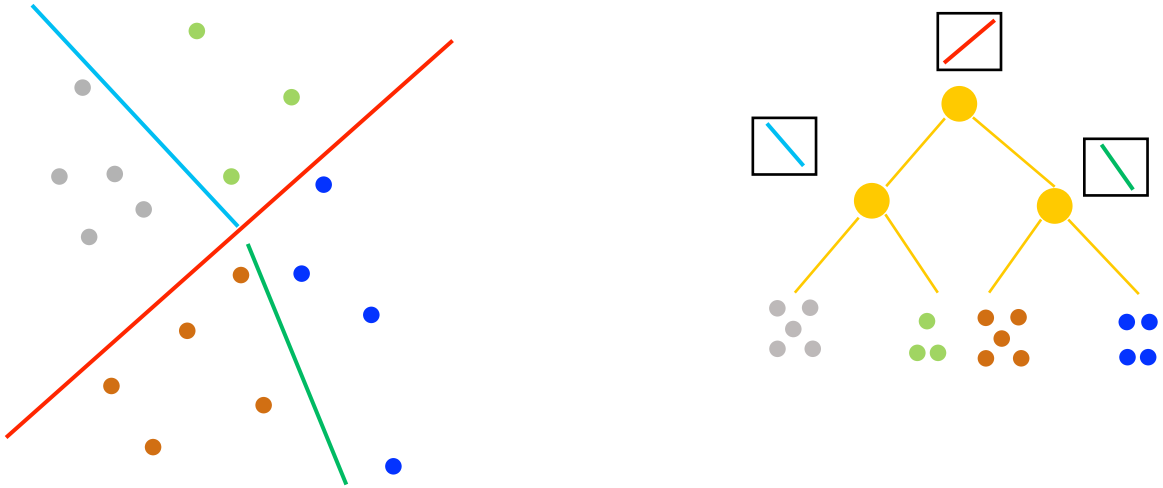


Random Projection Trees



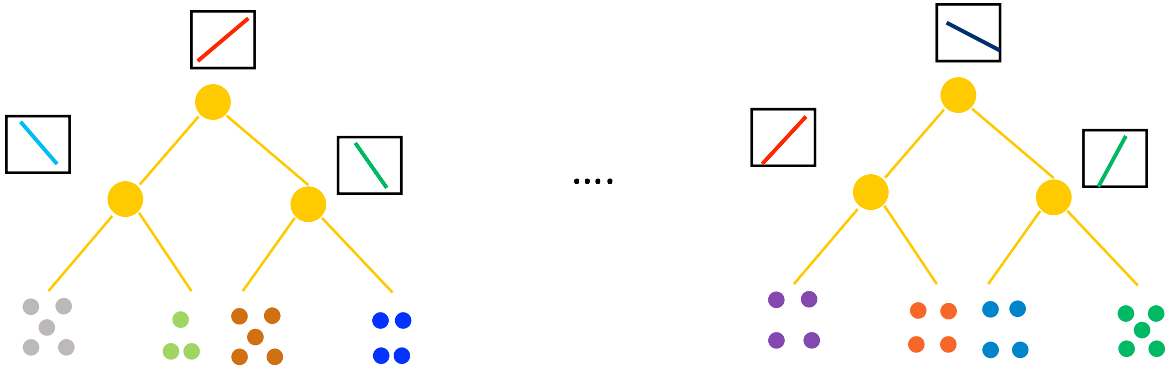


Random Projection Trees



K-NNG Construction

- Search nearest neighbors through traversing random projection trees
  - Only data points in the leaf are considered as nearest neighbors
- Multiple trees are usually used to improve the accuracy
  - e.g., hundreds



Reduce the Number of Trees

- Construct a less accurate K-NNG with *a few* trees
- Iteratively refine the K-NNG through "***neighbor exploring***"
  - "A neighbor of my neighbor is also likely to be my neighbor"
  - *Second-order* neighbors are also treated as candidates of *first-order* neighbors

Results of K-NNG Construction

- X axis: accuracy of K-NNG
  - With different values of parameters
- Y axis: running time (minutes)
- tSNE: 16 hours (95% accuracy)
- LargeVis: 25 minutes
  - ***>30*** times faster than t-SNE

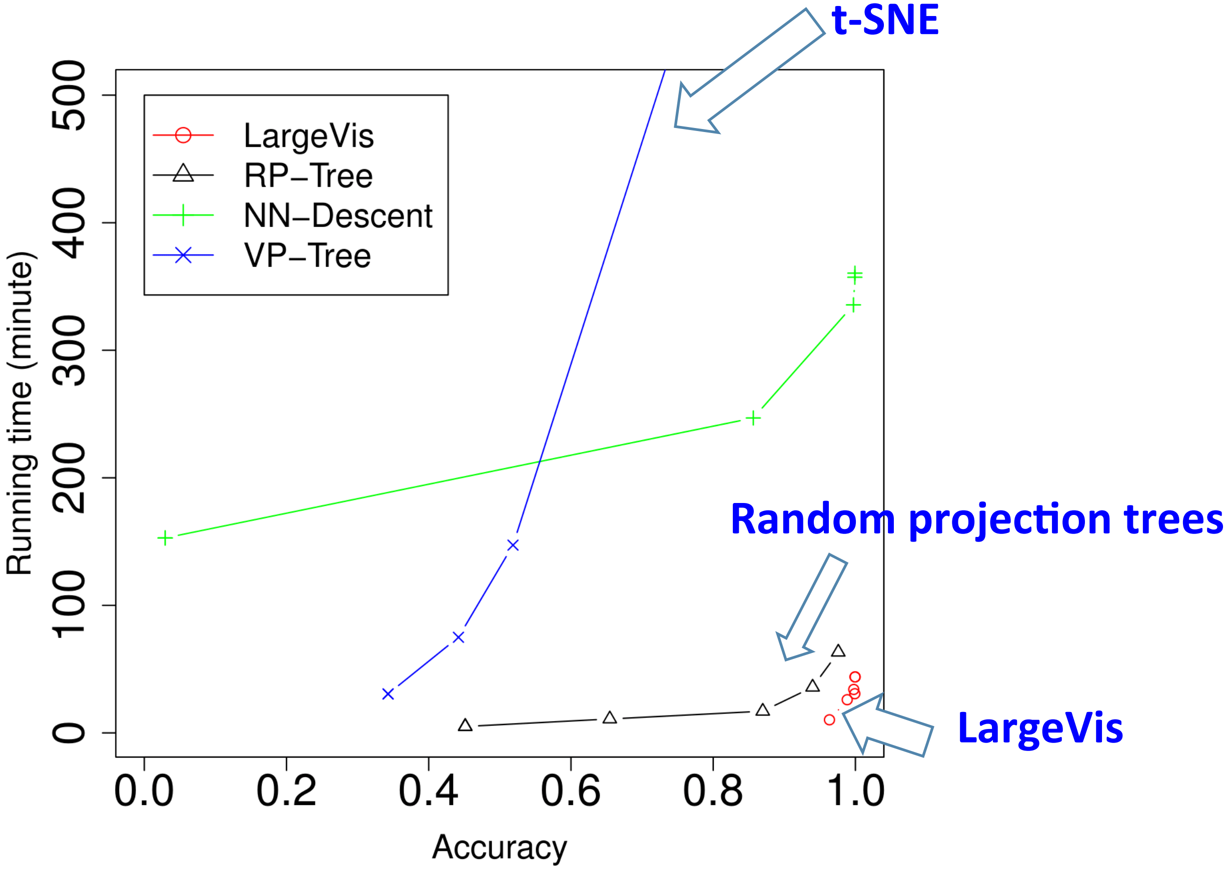


Fig.: Results on 3 Million Data with 100 Dimension

A Probabilistic Model for Graph Layout

- Preserve the similarities of the vertices in 2D/3D space
  - Represent each vertex *i* with a 2D/3D vector $\vec{y}_i$
  - Keep *similar* data close while *dissimilar* data far apart
- Probability of observing a *binary* edge between vertices *(i,j)*:

$$p(e_{ij}=1)=\frac{1}{1+\|\vec{y}_i-\vec{y}_j\|^2}$$

- Likelihood of observing a *weighted* edge between vertices (i,j):

$$p(e_{ij}=w_{ij})=p(e_{ij}=1)^{w_{ij}}$$

A Probabilistic Model for Graph Layout

- Objective:

$$O = \prod_{(i,j)\in E} p(e_{ij} = w_{ij}) \prod_{(i,j)\in \overline{E}} (1 - p(e_{ij} = 1))^{\gamma}$$

Weight of the negative edges

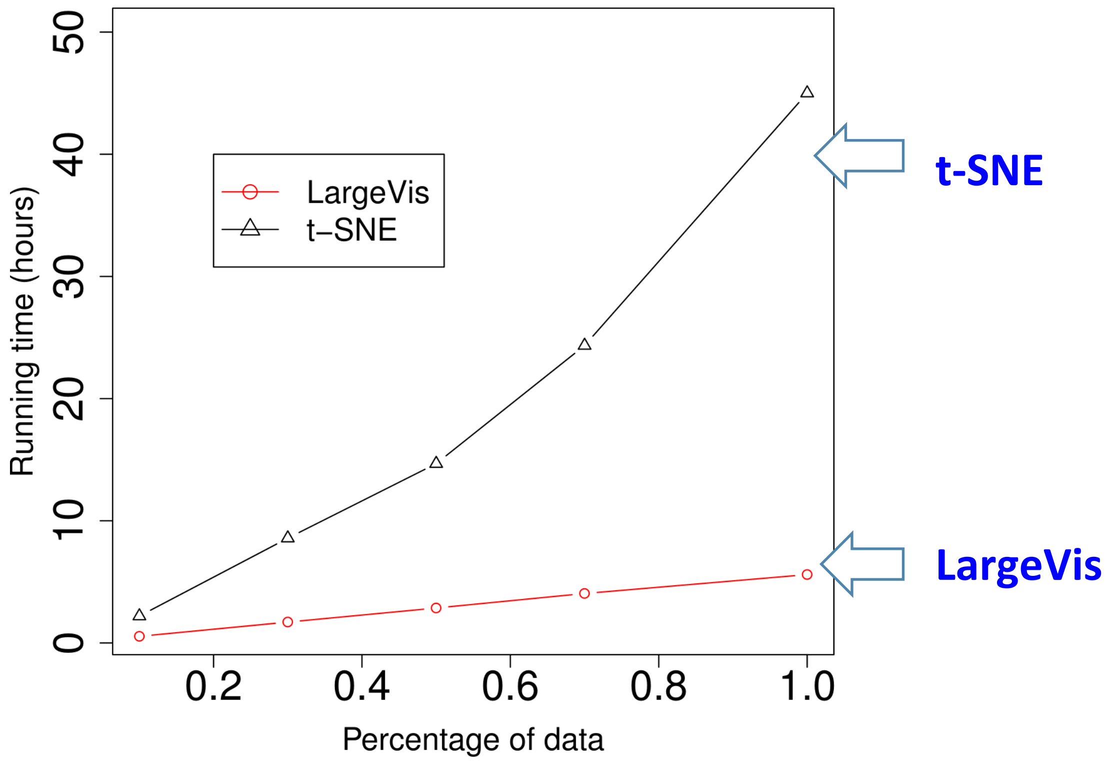
Positive edges

Negative edges

- Randomly sample some negative edges
- Optimized through asynchronous stochastic gradient descent
- Time complexity: ***linear*** to the number of data points

Efficiency of Graph Layout

- Time complexity
  - t-SNE: O(NlogN)
  - LargeVis: O(N)
- On 3 million data points
  - t-SNE: 45 hours
  - LargeVis: 5.6 hours
  - *Seven* times faster



Visualization Quality

- Metric: *classification accuracy* with KNN on 2D space
- Configuration:
  - LargeVis with *default* parameters
  - t-SNE with *default* and *optimal* parameters (tuned per data set)
- LargeVis ≈ tSNE with optimal parameters
- LargeVis >> tSNE with default parameters
- Parameters of LargeVis are very *stable*

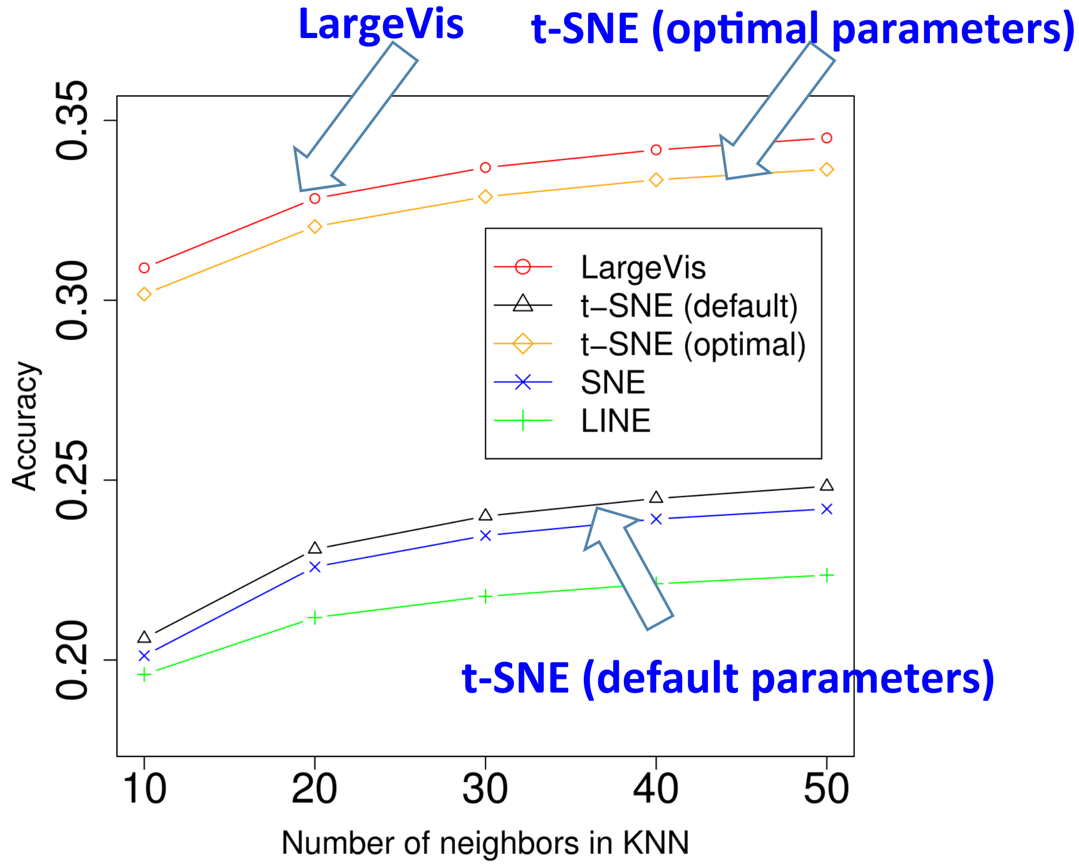
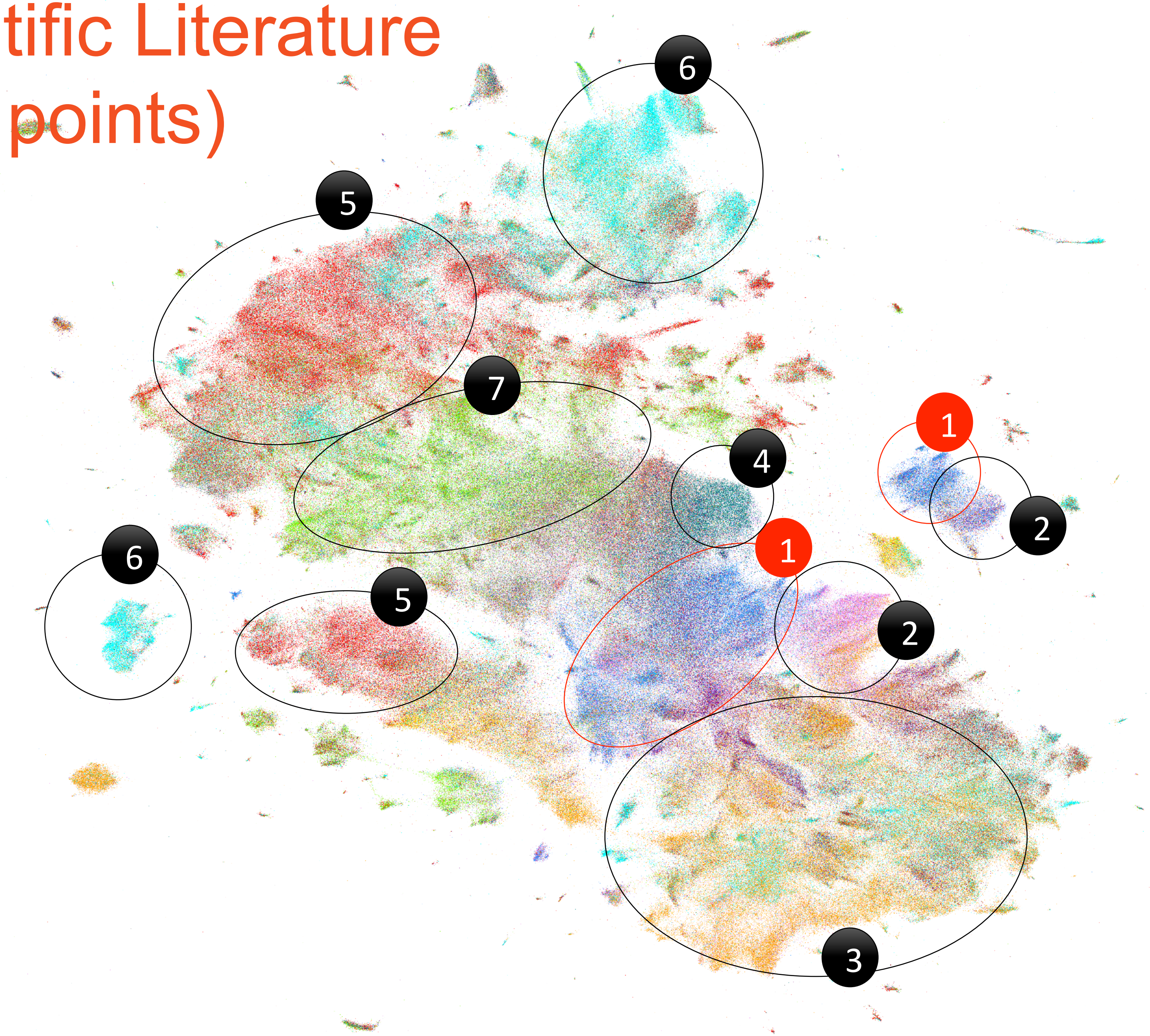


Fig.: Results on 3 Million Data with 100 Dimension

Take Away

- **LargeVis**: a new technique for big data layout
- Efficient K-nearest neighbor graph construction
  - Random projection trees + neighbor exploring
- Efficient and effective probabilistic model for graph layout
  - Complexity ***linear*** to the number of data points
  - ***Stable*** parameters
- The layout computed by LargeVis can facilitate many visualizations.
- We will release the source code very soon!

Scientific Literature (10M points)



1 Computer Science

2 Mathematics

3 Physics

4 Economics

5 Biology

6 Chemistry

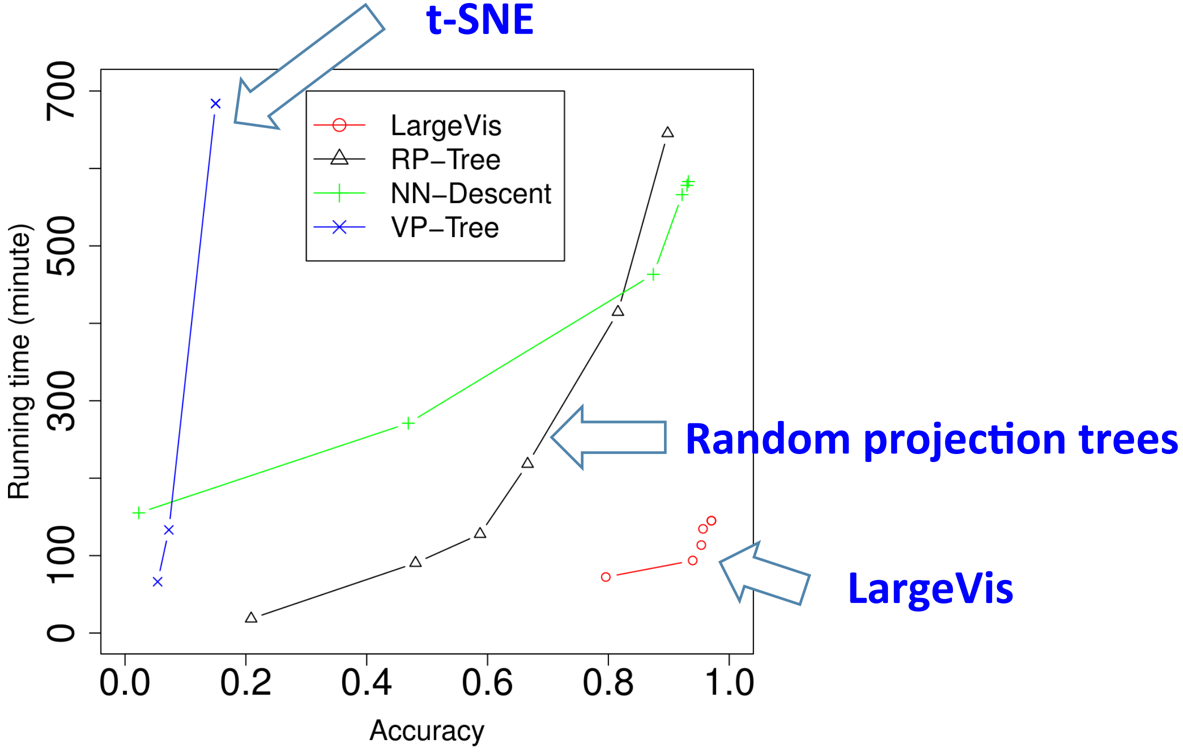
7 Medicine

Running Time (hours) of **t-SNE** and **LargeVis**

	~1M	~3M	~4M	~2M	~1.5M
Dataset	**WikiWord**	**WikiDoc**	**LiveJournal**	**CSAuthor**	**DBLPPaper**
t-SNE	9.82	45.01	70.35	28.33	18.73
LargeVis	2.01	5.60	9.26	4.24	3.19
Speedup Rate	3.9	**7**	**6.6**	5.7	4.9

Construct K-NNG on 3 Million Data with 100 Dimension

- X axis: accuracy of K-NNG
  - With different values of parameters
- Y axis: running time (minutes)
- LargeVis: ~ 2 hours
- Very hard to yield a very accurate K-NNG with random projection trees


t-SNE
Random projection trees
LargeVis
LargeVis
RP−Tree
NN−Descent
VP−Tree
Running time (minute)
Accuracy