20210929 Hae Uma WAN Zin DEDA Dim Reduction VizTech Unit_12.key

Dim Reduction VizTech

Elizaveta Zinovyeva

Lucas Umann

Bingling WANG

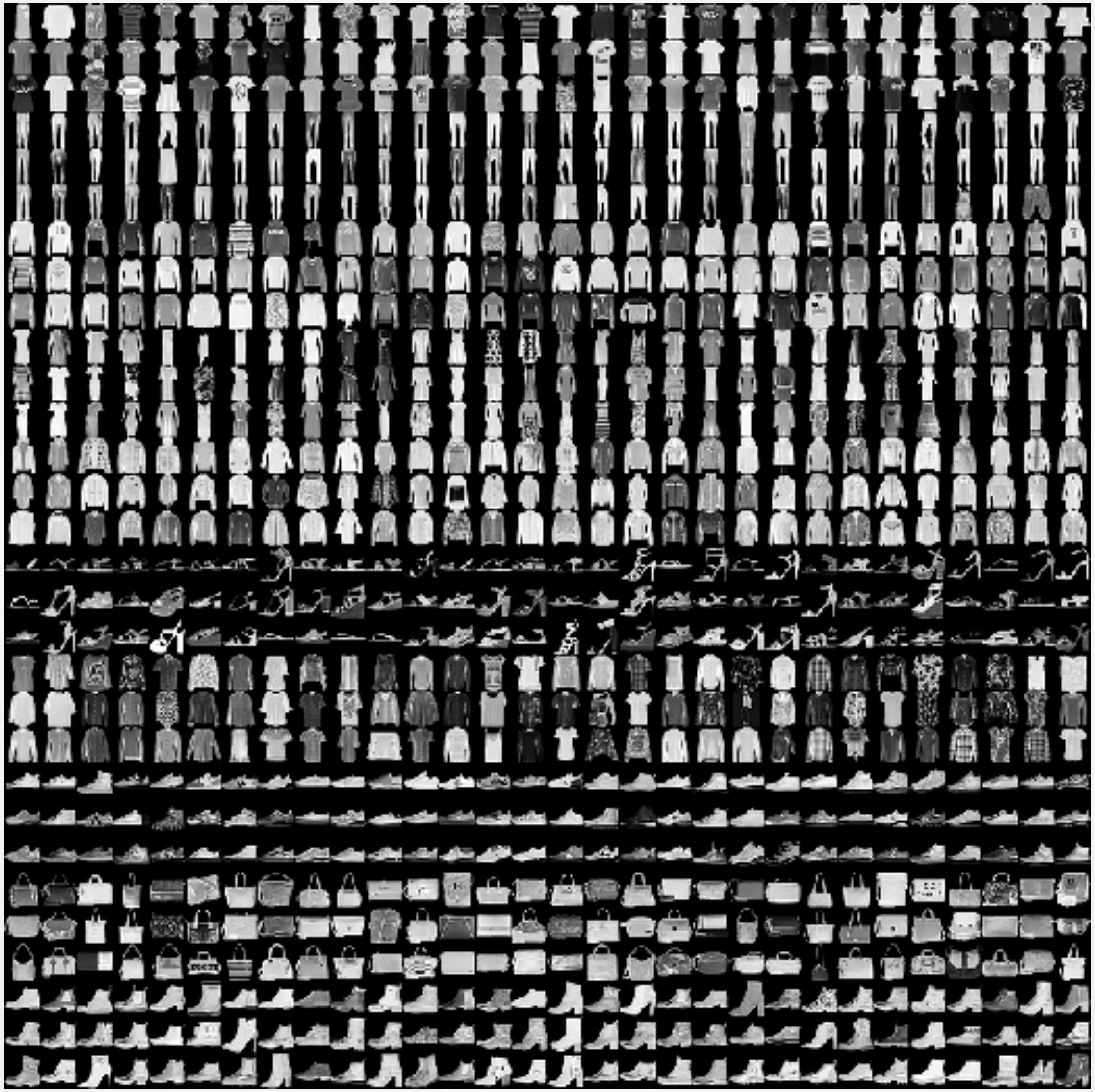
Wolfgang Karl Härdle

Ladislaus von Bortkiewicz Professor of Statistics

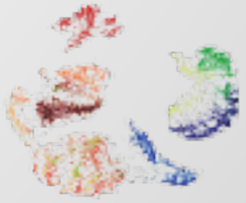
Humboldt-Universität zu Berlin

lvb.wiwi.hu-berlin.de

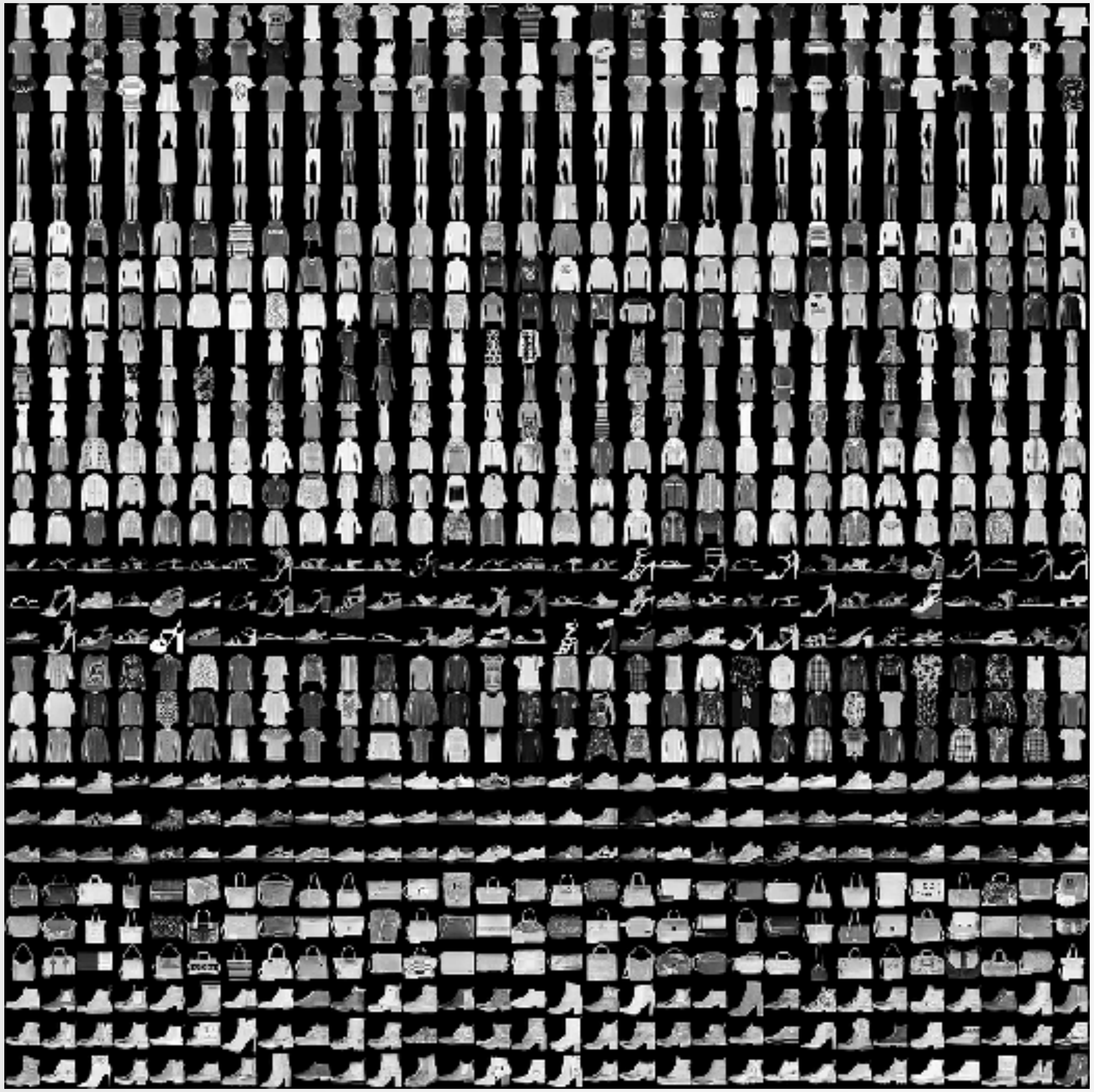
Fashion-MNIST



Fashion MNIST dataset
Source: Zalando Research



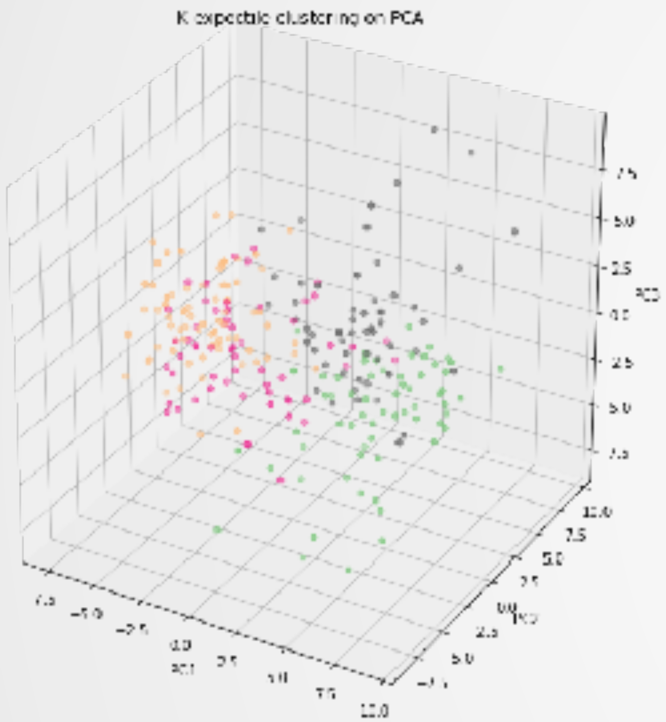
Fashion-MNIST

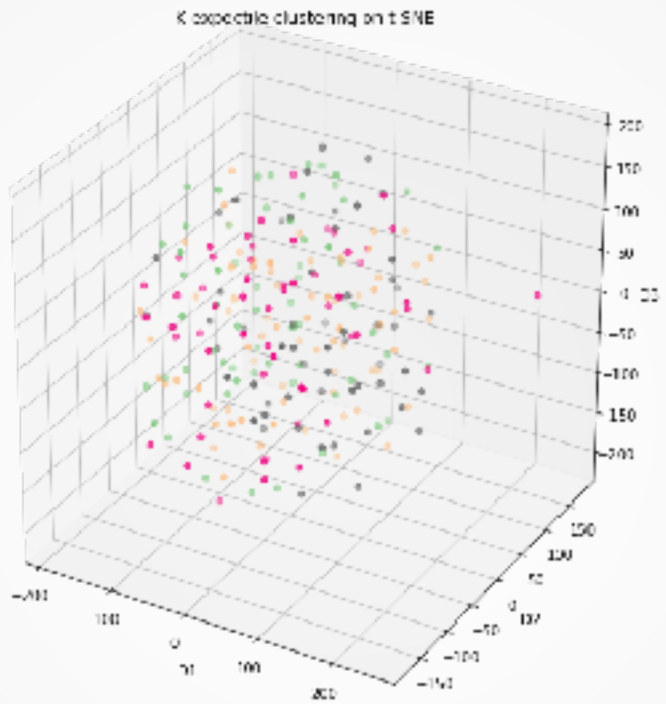


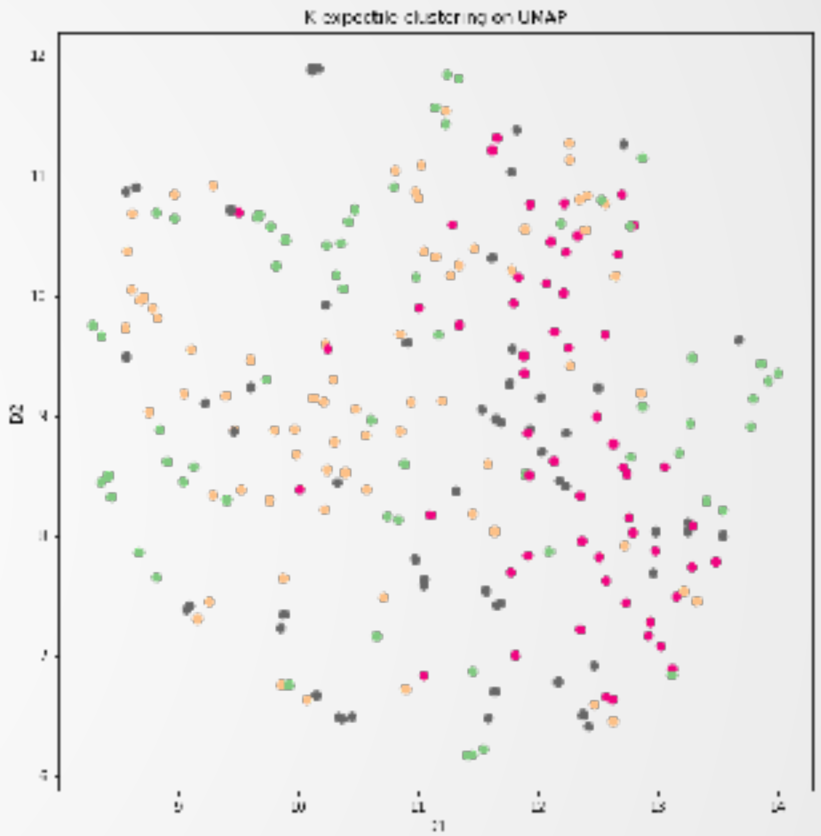
Fashion MNIST dataset
Source: Zalando Research

- Zalando article images
- 60.000 training, 10.000 test
- 28x28 grayscale
- dim = 784
- No interpretability of raw data by human observer

Classification





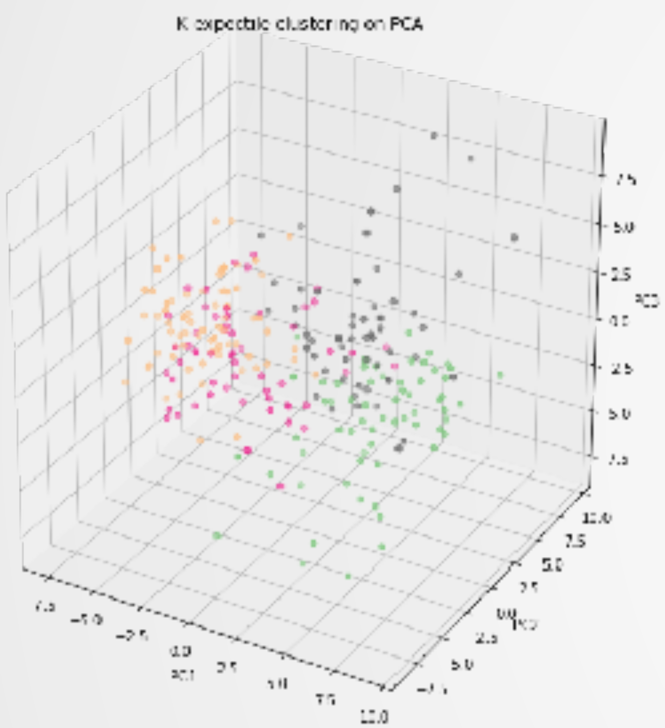


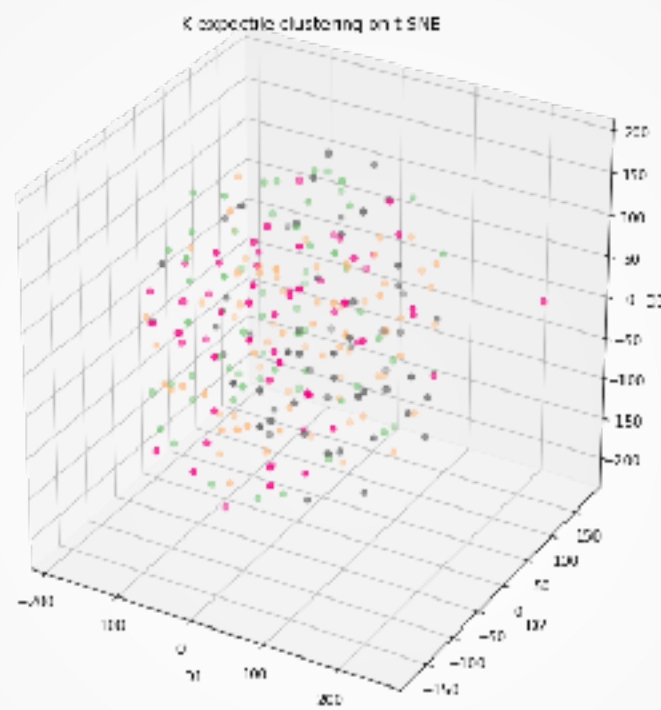
n_neighbors=5, min_dist =0.1

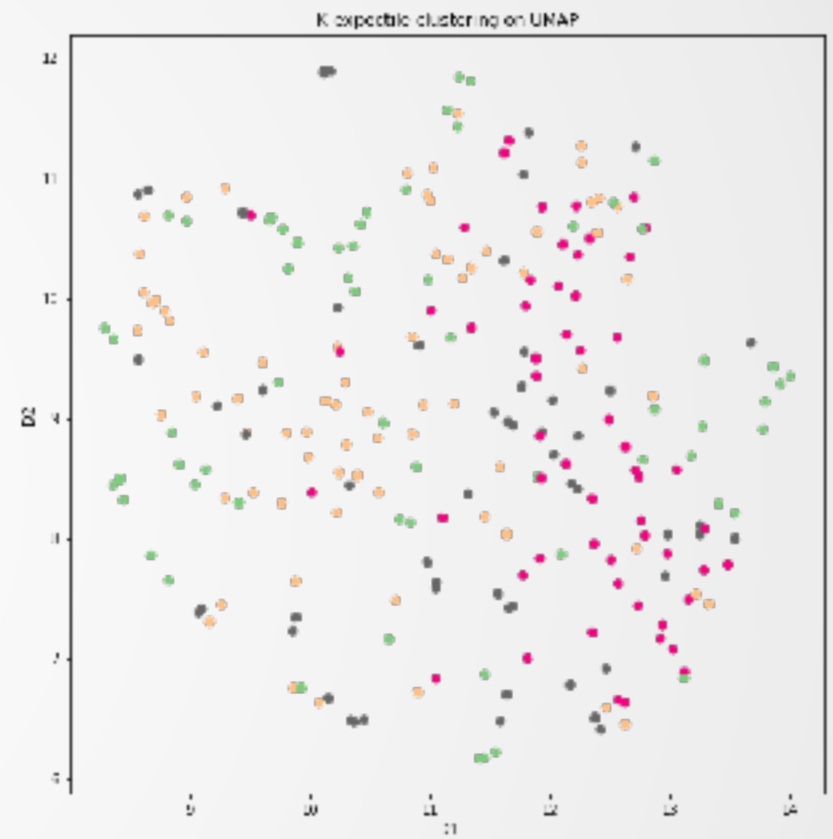
Data Source: CoinGecko crypto returns 20190730-20200117

248 observations, 998 dimensions reduced to 3 UMAP

Classification







n_neighbors=5, min_dist =0.1

- Pattern similarities between crypto returns are hard to describe in high dim
- Dim reduction helps to describe them in different ways

UMAP

CryptoPunk

CryptoPunk Images
Source: larvalabs.com

- Cryptopunk a NFT non-fungible token
- Are there clusters in high dim?
- Dim reduction provides good start

Clustering in High Dimensions: 4 problems

Curse of dimensionality

Clustering in High Dimensions: 4 problems

Curse of dimensionality

Distances less precise:

$$\lim_{dim \to \infty} \frac{dist_{max} - dist_{min}}{dist_{min}} = 0$$

Clustering in High Dimensions: 4 problems

Distances less precise:

$$\lim_{dim \to \infty} \frac{dist_{max} - dist_{min}}{dist_{min}} = 0$$

$dist_{min}/dist_{max}$ min/max distance

between all data points

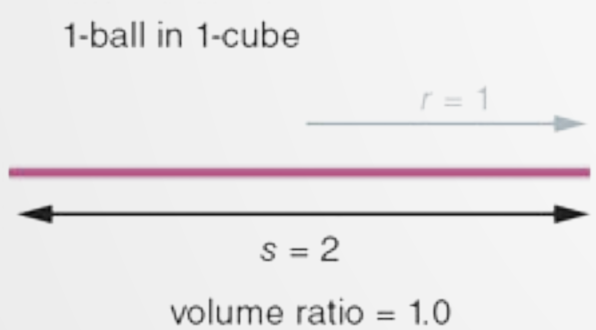
Clustering in High Dimensions: 4 problems

Distances less precise:

$$\lim_{dim \to \infty} \frac{dist_{max} - dist_{min}}{dist_{min}} = 0$$

$dist_{min}/dist_{max}$ min/max distance

between all data points



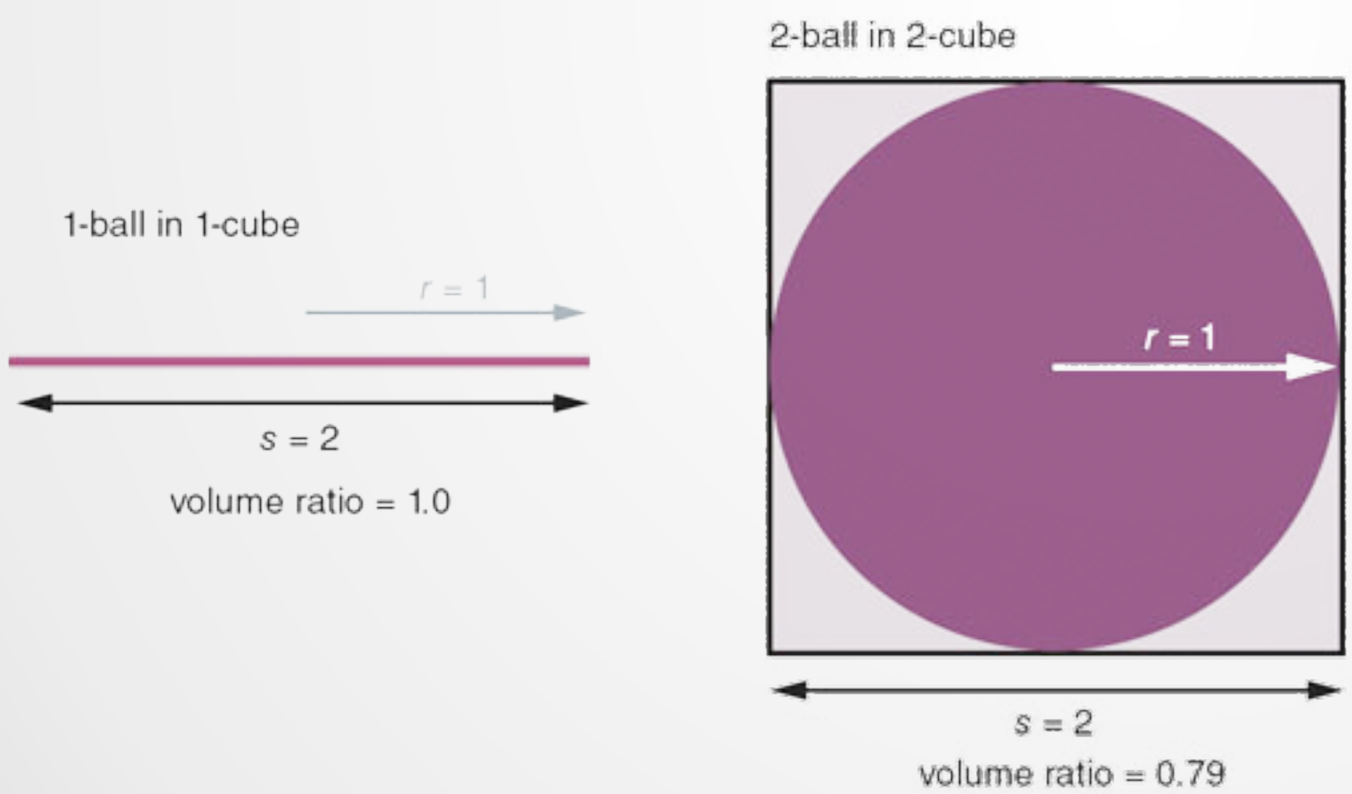
Clustering in High Dimensions: 4 problems

Distances less precise:

$$\lim_{dim \to \infty} \frac{dist_{max} - dist_{min}}{dist_{min}} = 0$$

$dist_{min}/dist_{max}$ min/max distance

between all data points



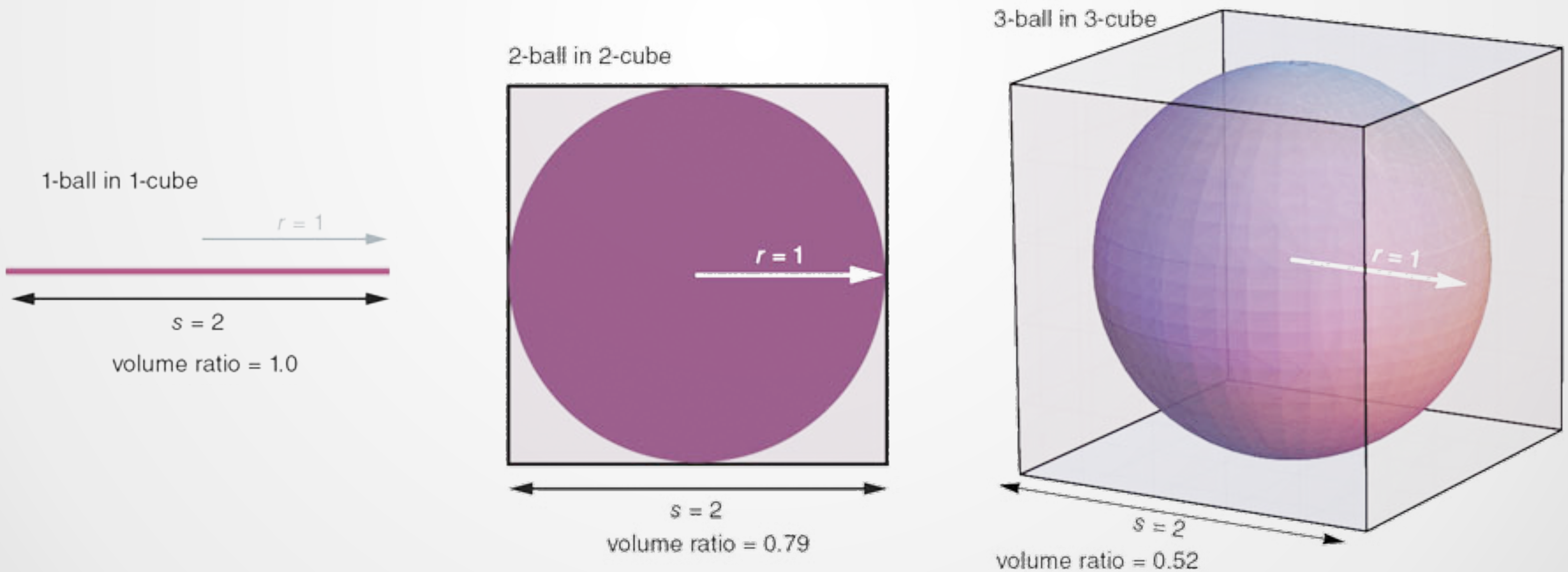
Clustering in High Dimensions: 4 problems

Distances less precise:

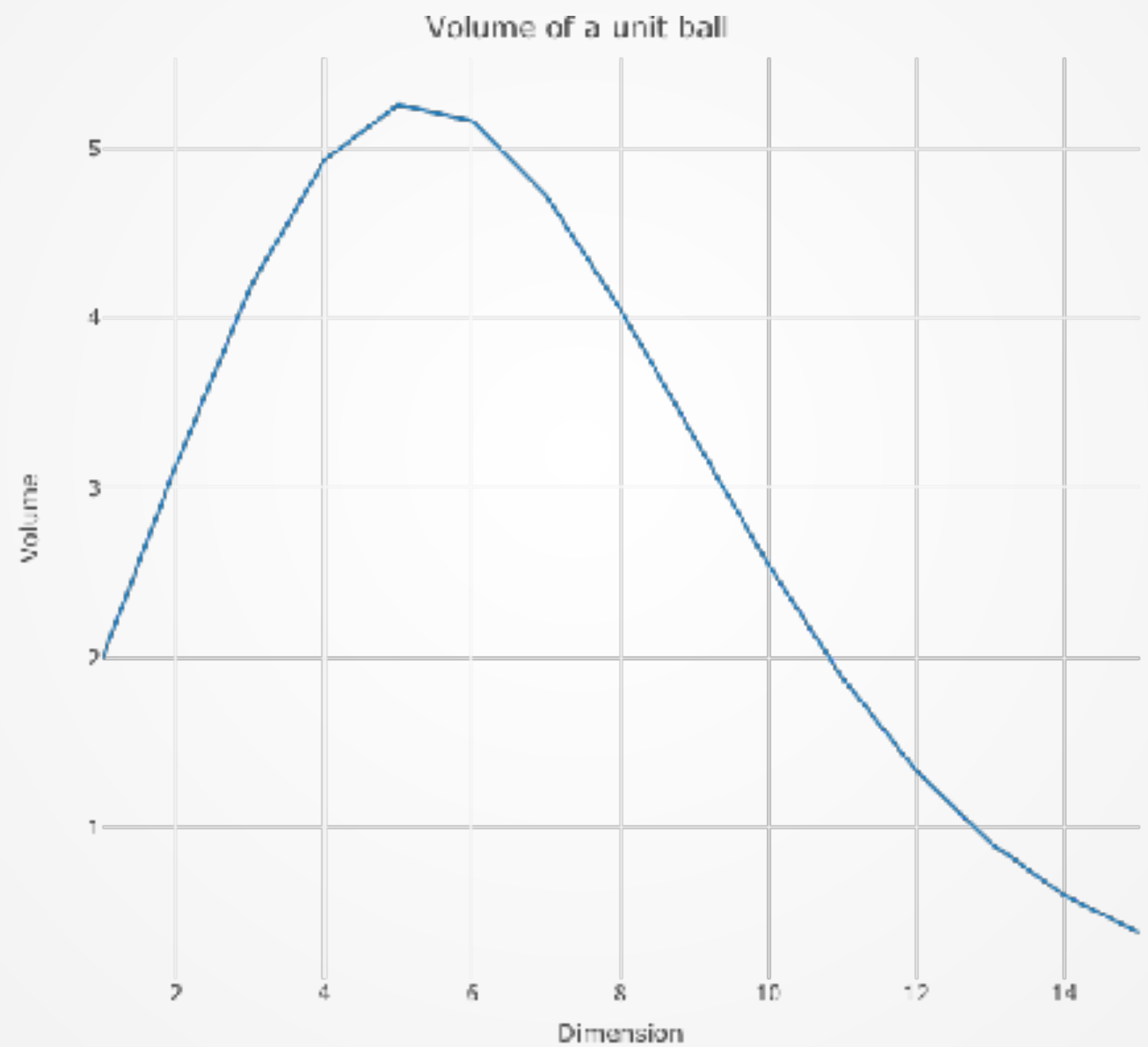
$$\lim_{dim \to \infty} \frac{dist_{max} - dist_{min}}{dist_{min}} = 0$$

$dist_{min}/dist_{max}$ min/max distance

between all data points

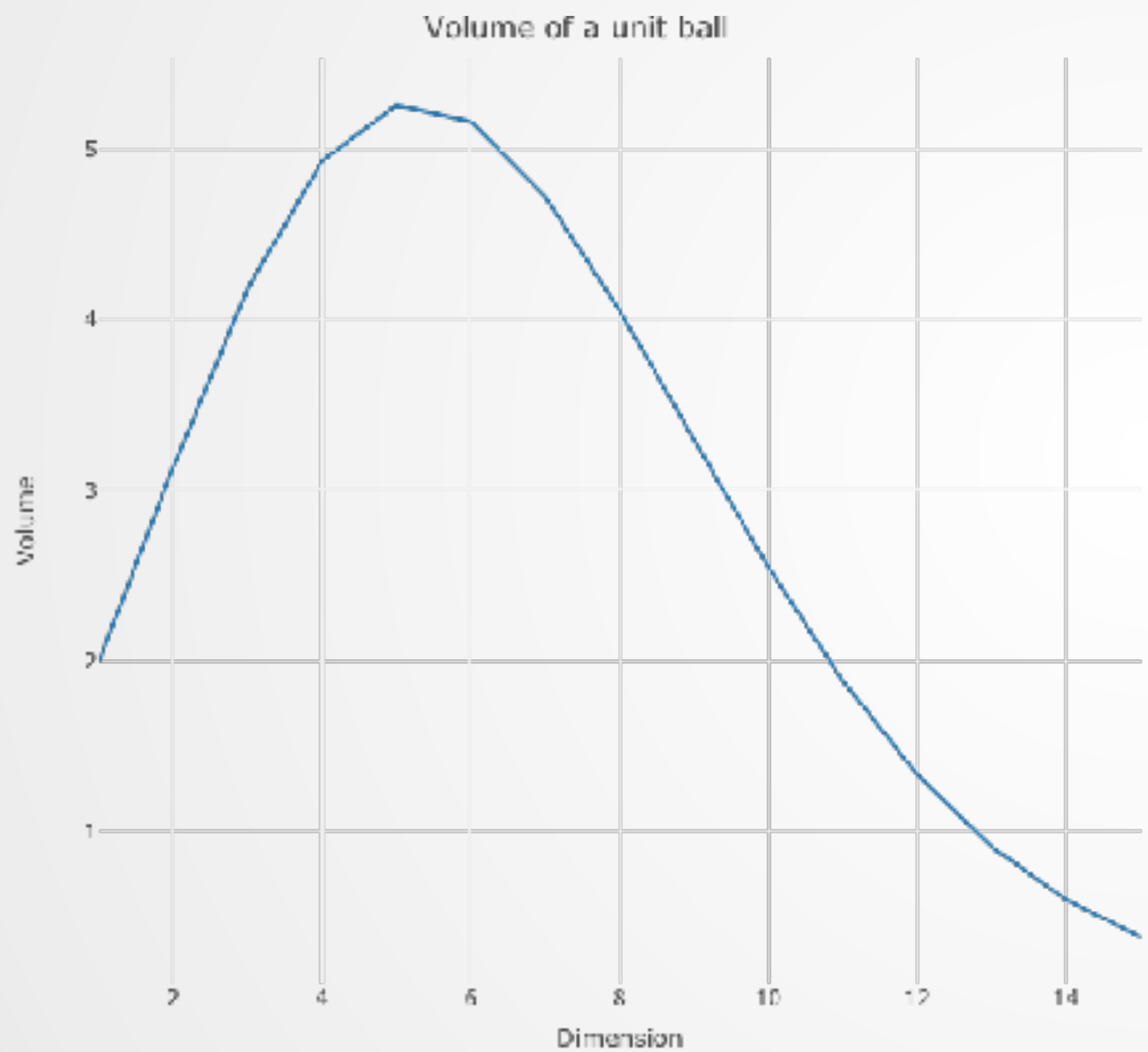


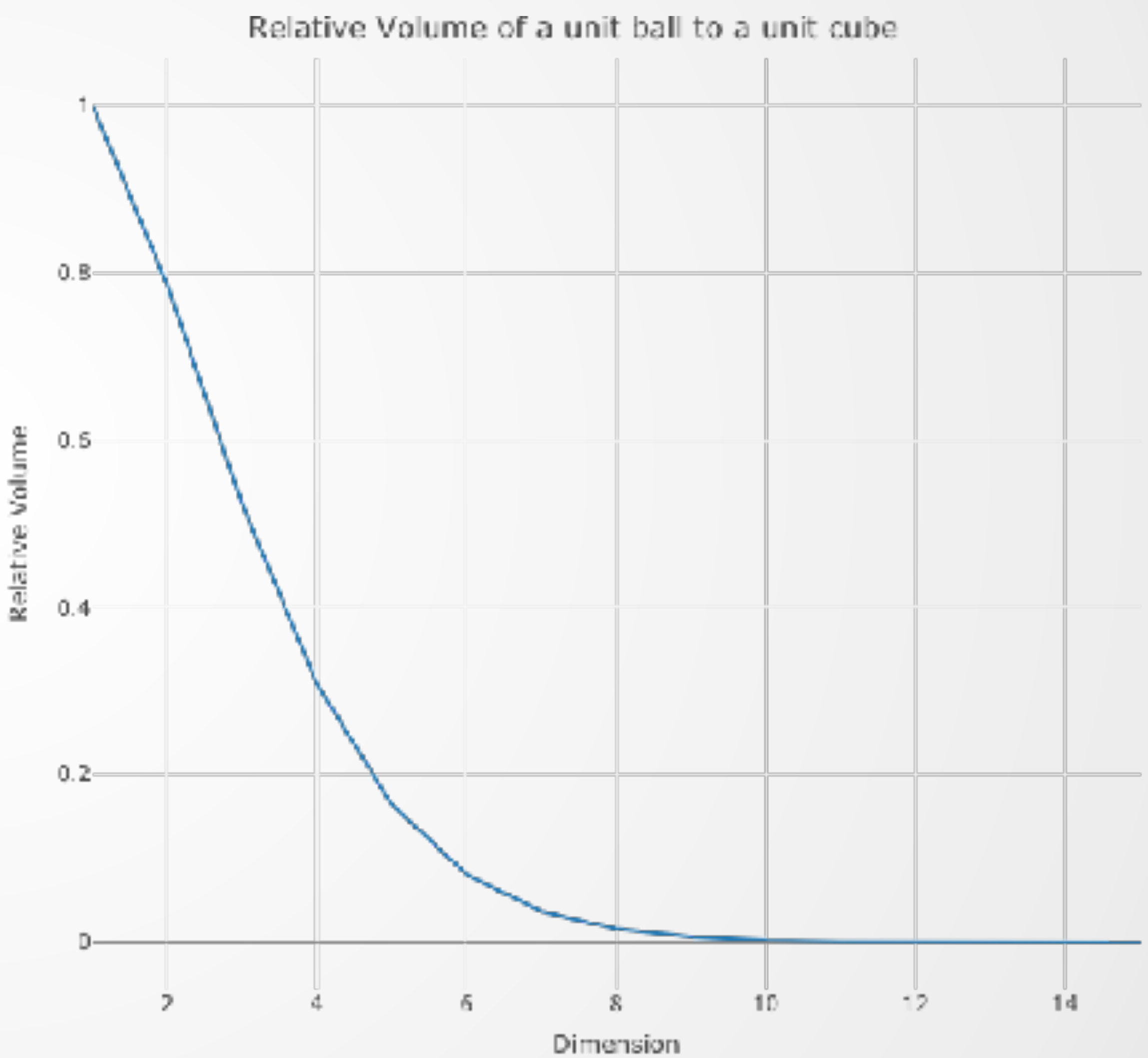
Clustering in High Dimensions: 4 problems



Volume of a d-dimensional Unit Ball: $V_d = \dfrac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2} + 1)}$

Clustering in High Dimensions: 4 problems

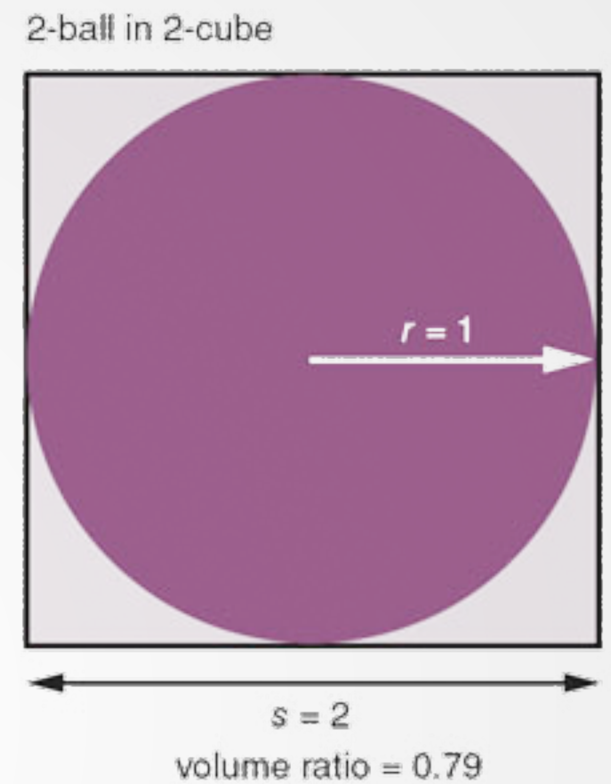


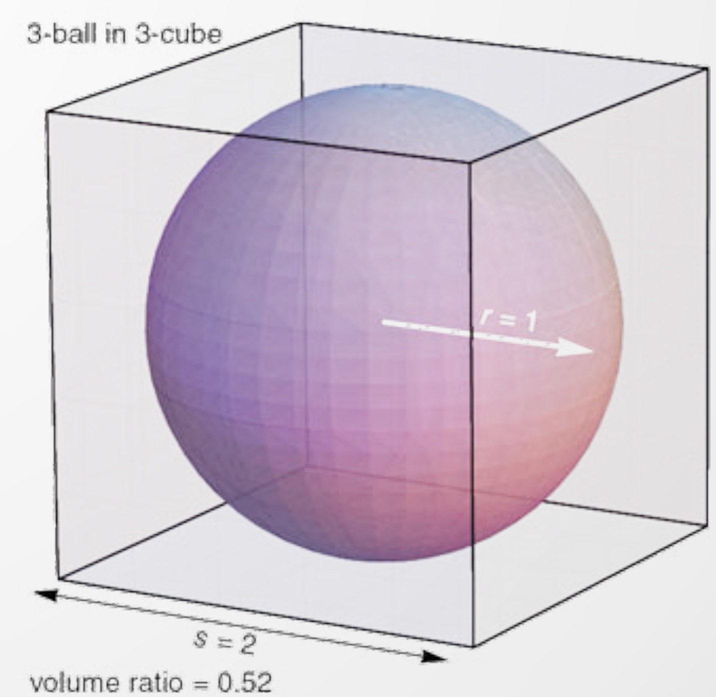


Volume of a d-dimensional Unit Cube (side length = 2): $V_d = 2^d$

Clustering in High Dimensions: 4 problems

- Assume $n = 100$ data points uniformly distributed in unit cube
- Consider unit ball around middle point
- 2 dim.: 79 points inside ball
- 5 dim.: 16 points
- 15 dim.: need 100000 data points to get 1 inside





Clustering in High Dimensions: 4 problems

Curse of dimensionality

Distances less precise:

$$\lim_{dim \to \infty} \frac{dist_{max} - dist_{min}}{dist_{min}} = 0$$

Clustering in High Dimensions: 4 problems

Curse of dimensionality

Distances less precise:

$$\lim_{dim \to \infty} \frac{dist_{max} - dist_{min}}{dist_{min}} = 0$$

High number of attributes: many may not be meaningful for a given cluster

Clustering in High Dimensions: 4 problems

Curse of dimensionality

Distances less precise:

$$\lim_{dim \to \infty} \frac{dist_{max} - dist_{min}}{dist_{min}} = 0$$

High number of attributes: many may not be meaningful for a given cluster

Likely that some attributes are correlated

Clustering in High Dimensions: 4 problems

Curse of dimensionality

Distances less precise:

$$\lim_{dim \to \infty} \frac{dist_{max} - dist_{min}}{dist_{min}} = 0$$

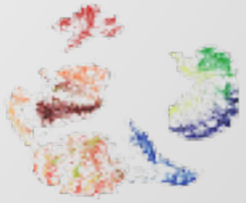
High number of attributes: many may not be meaningful for a given cluster

Likely that some attributes are correlated

Dim reduction increases accuracy. VizTech to the rescue!

Outline

1. Motivation
2. *t*-distributed Stochastic Neighbor Embedding (*t*-SNE)
3. *t*-SNE example
4. Uniform Manifold Approximation and Projection (UMAP)
5. UMAP example
6. *t*-SNE vs. UMAP



t-SNE VizTech

- *t*-SNE *t*-distributed stochastic neighbor embedding
- Non-linear VizTech with preservation of local structure
- Embedding: preserve as much *significant structure* of high dim data as *possible* in a low dim

$$X = \{x_1, x_2, \ldots, x_n\} \longrightarrow Y = \{y_1, y_2, \ldots, y_n\}$$

- x_i in $\mathbb{R}^{\mathrm{k}}$, y_i in $\mathbb{R}^{\mathrm{l}}$, $k > l$

t-SNE of Fashion MNIST. Source: Researchgate

t-SNE VizTech

- Transform high dim (HD) distances into conditional probabilities:

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k\neq i} \exp(-\|x_i - x_k\|^2/2\sigma_i^2)}$$

  - shows how close is the point x_j to the point x_i with Gaussian distribution around x_i with deviation σ_i

t-SNE VizTech

- Transform high dim (HD) distances into conditional probabilities:

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2/2\sigma_i^2)}$$

  - shows how close is the point x_j to the point x_i with Gaussian distribution around x_i with deviation σ_i
- σ_i must be chosen for each data point individually.
  - it is chosen such that points in dense areas are given a smaller variance than points in sparse areas.
  - σ_i induces a probability distribution P_i
  - *t*-SNE performs binary search for the value of σ_i that produces P_i with fixed perplexity specified by a user

$$Perp(P_i) = 2^{H(P_i)}, \quad H(P_i) = -\sum_j p_{j|i} \log_2\{p_{j|i}\}$$

t-SNE VizTech

- Symmetrized version of conditional similarity in HD:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

t-SNE VizTech

- Symmetrized version of conditional similarity in HD:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

- Similarity in the low dim (LD) space:
  - *t*-student with 1 degree of freedom

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}}$$

t-SNE VizTech

- Symmetrized version of conditional similarity in HD:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

- Similarity in the low dim (LD) space:
  - *t*-student with 1 degree of freedom

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}}$$

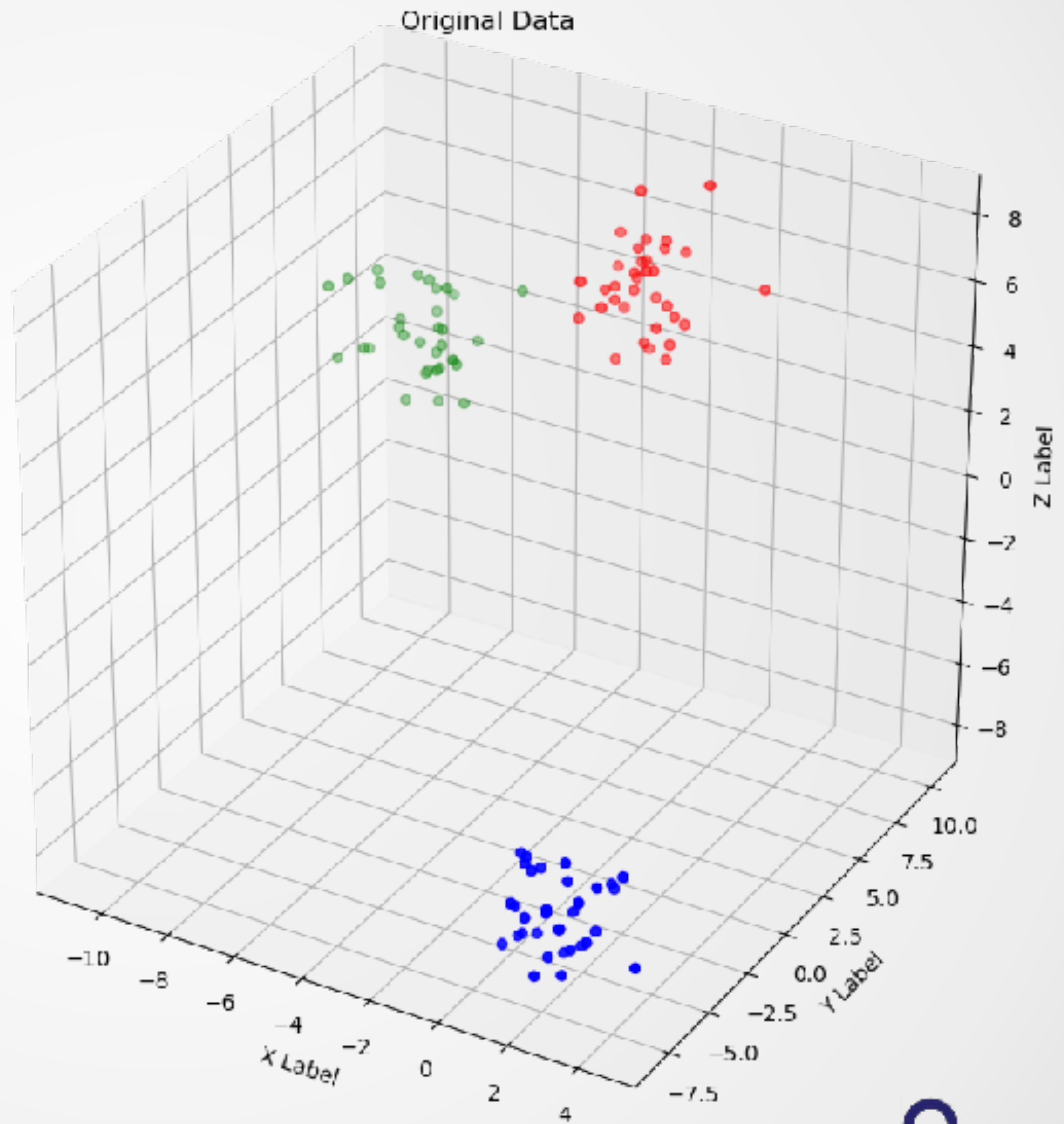
- Minimise Kullback-Leibler divergence using gradient descent

$$C = KL(P\,||\,Q) = \sum_i \sum_j p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right)$$

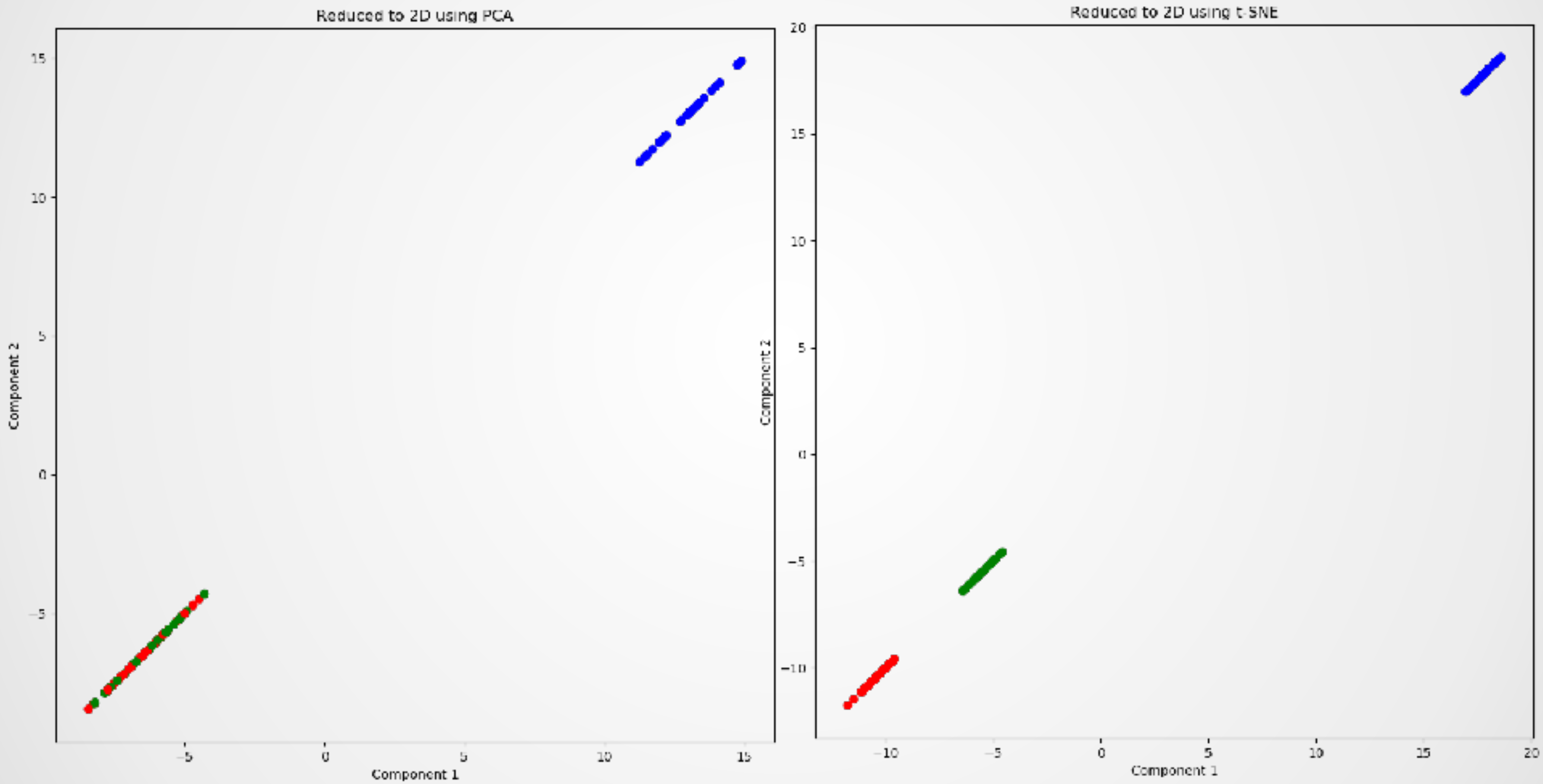
Kullback, Leibler in BBI

t-SNE vs PCA

- Original 3D data
  - 3 clusters
  - separable in 3D space
  - *t*-SNE aims to preserve local structure



t-SNE vs PCA

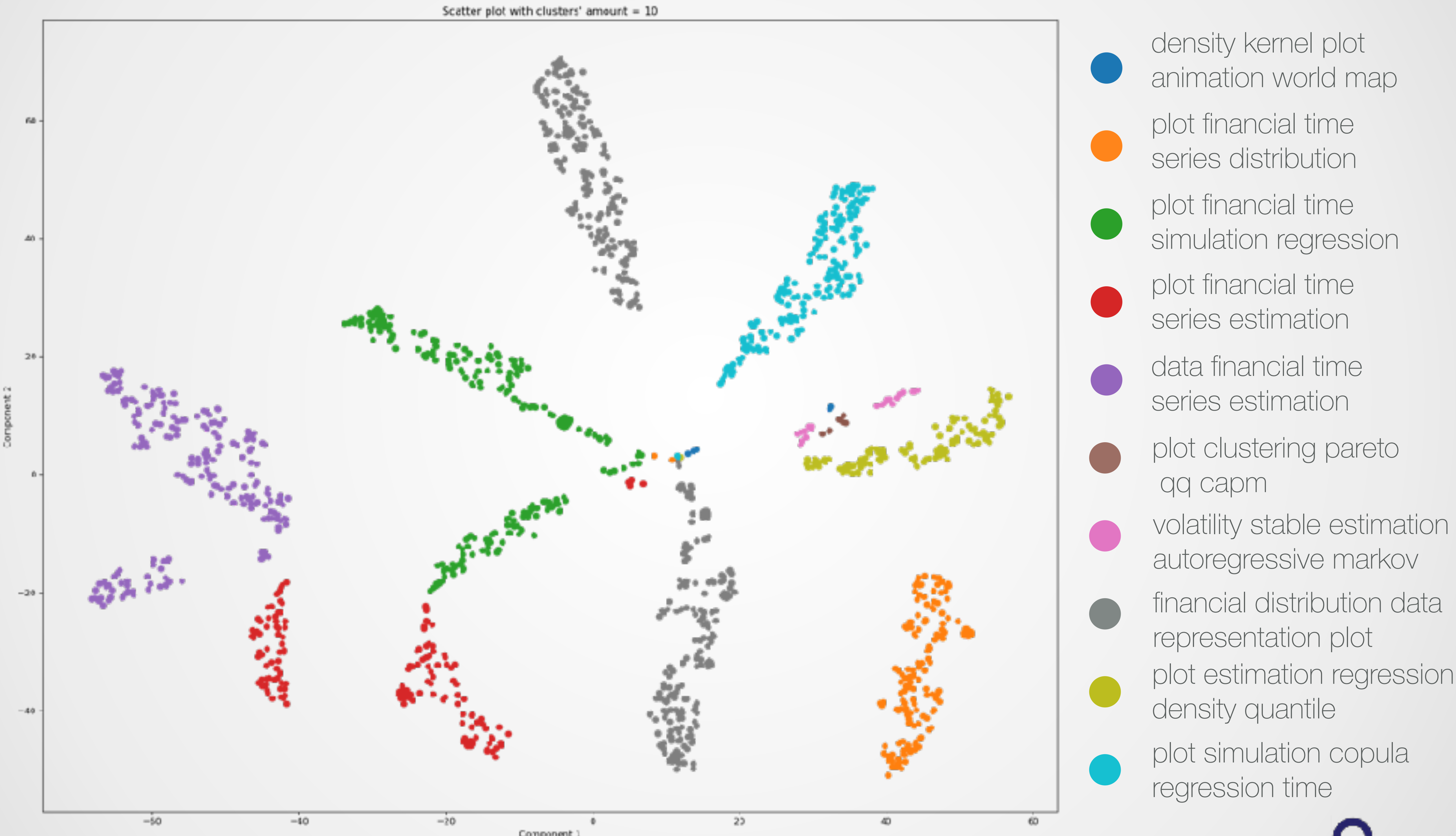

Reduced to 2D using PCA
Component 1
Component 2
Reduced to 2D using t-SNE
Component 1
Component 2

t-SNE algorithm

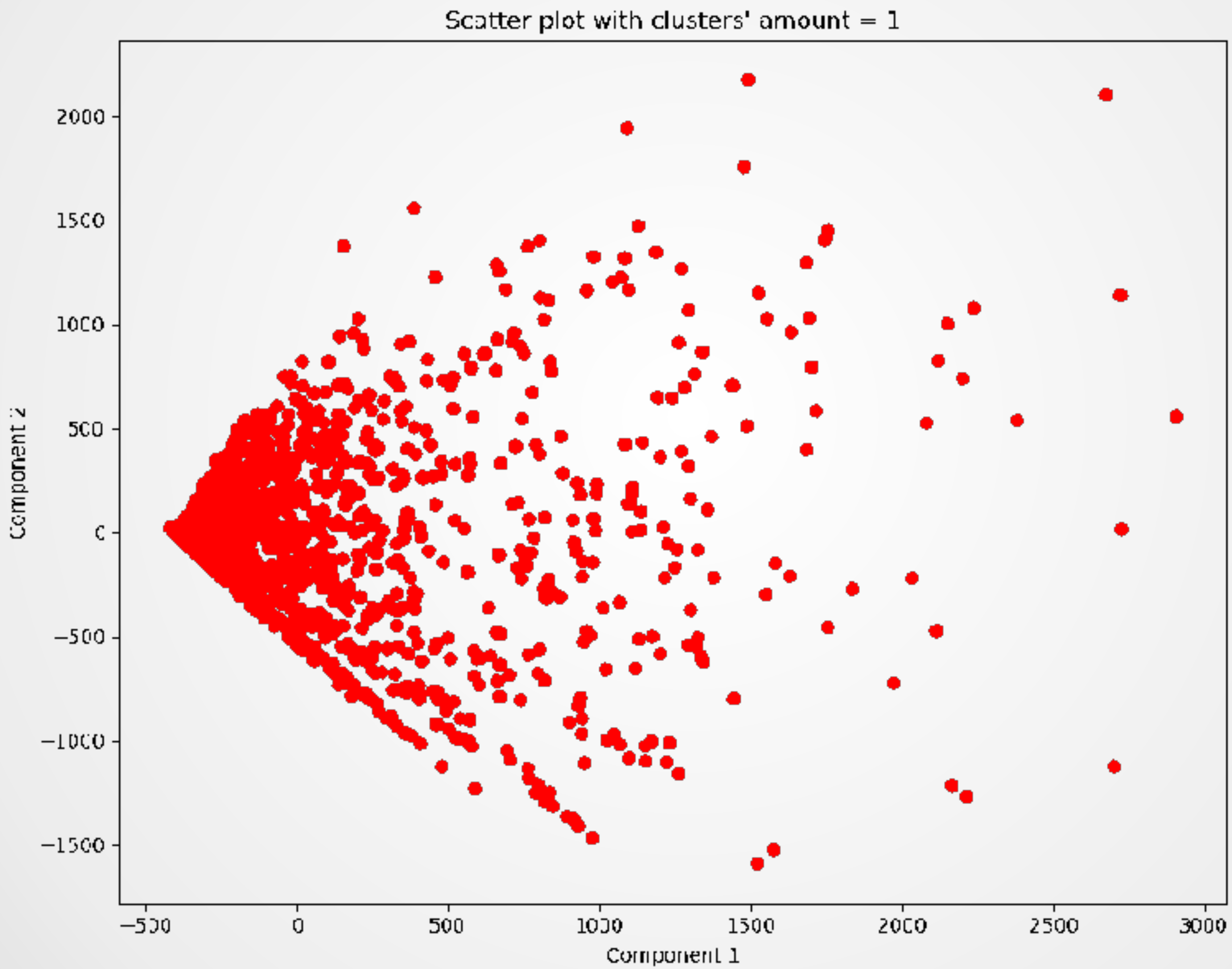
Algorithm 1: Simple version of t-Distributed Stochastic Neighbor Embedding.

Data: data set $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$,
cost function parameters: perplexity $Perp$,
optimization parameters: number of iterations T, learning rate η, momentum $\alpha(t)$.
Result: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, \ldots, y_n\}$.
begin
  compute pairwise affinities $p_{j|i}$ with perplexity $Perp$ (using Equation 1)
  set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$
  sample initial solution $\mathcal{Y}^{(0)} = \{y_1, y_2, \ldots, y_n\}$ from $\mathcal{N}(0, 10^{-4} I)$
  for $t = 1$ **to** T **do**
    compute low-dimensional affinities q_{ij} (using Equation 4)
    compute gradient $\frac{\delta C}{\delta \mathcal{Y}}$ (using Equation 5)
    set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t) \left(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)} \right)$
  end
end

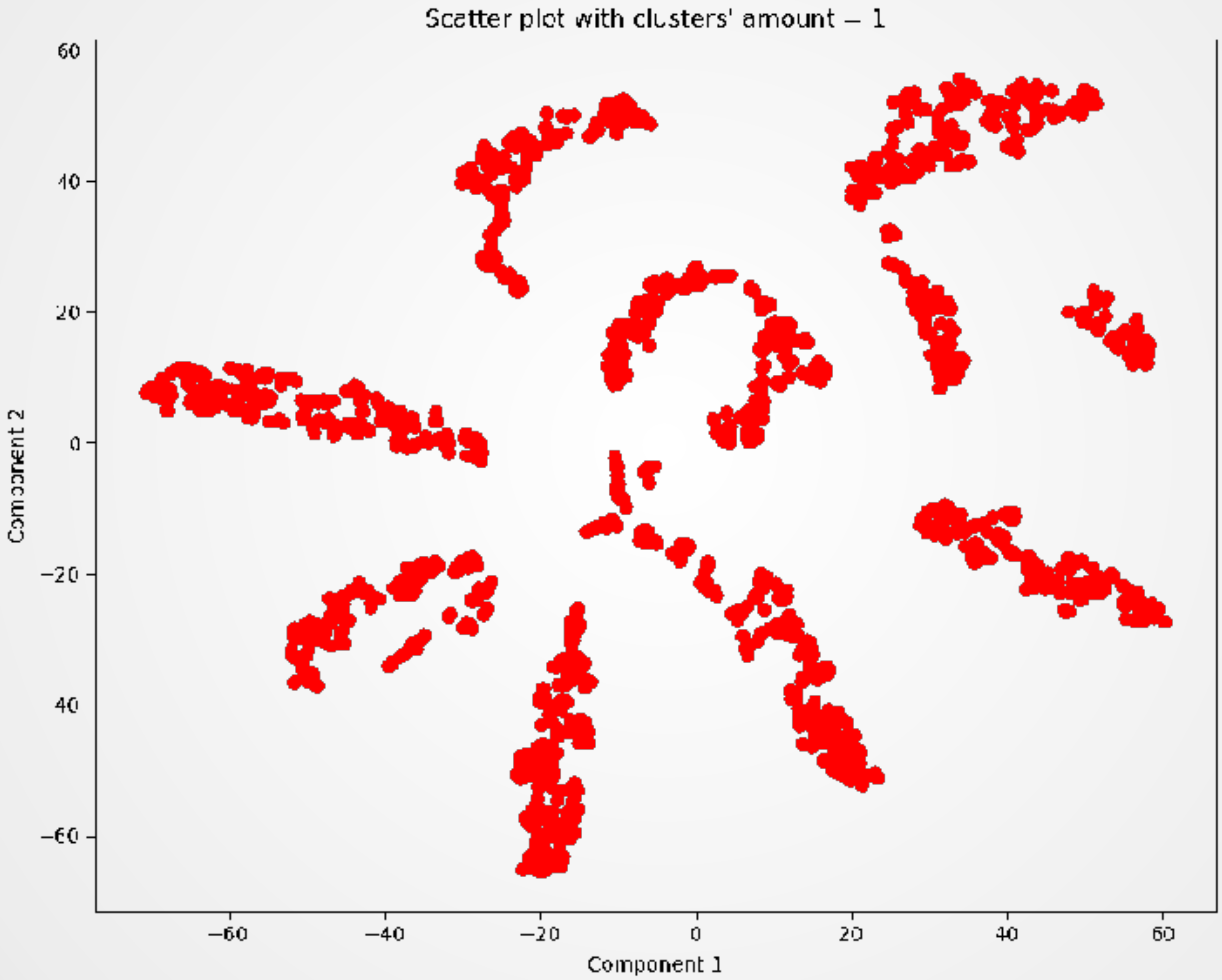
t-SNE VizTech



Dynamics of different amount of clusters (PCA)



Dynamics of different amount of clusters (*t*-SNE)

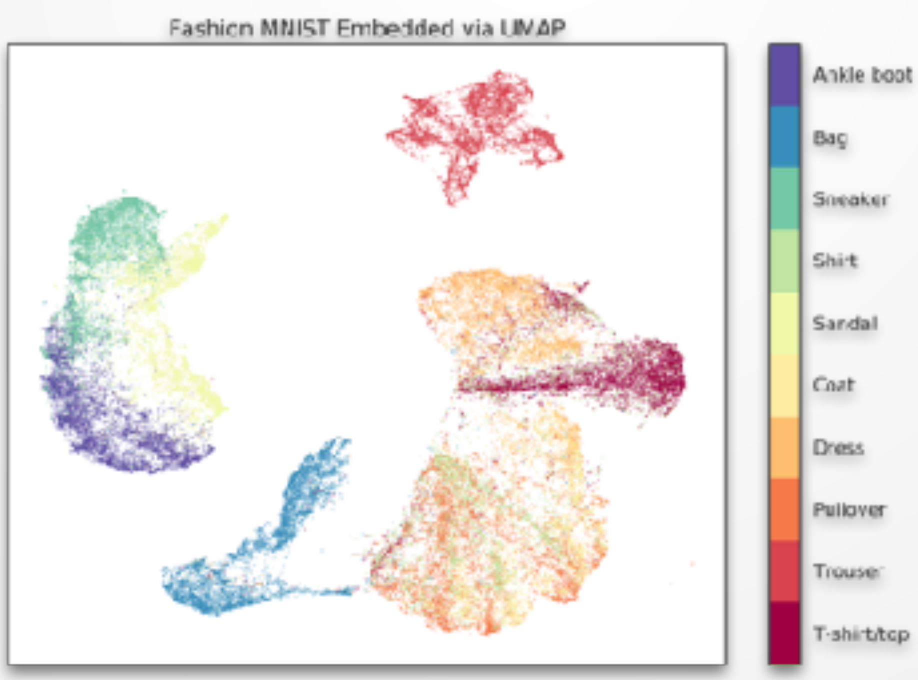


t-SNE bottomline

- Useful if the underlying application has a taxonomy.
- Agglomerative hierarchical clustering algorithms are expensive in terms of their computational and storage requirements.
- Merges are final and cannot be undone later, preventing global optimisation and causing trouble for noisy, high-dimensional data.

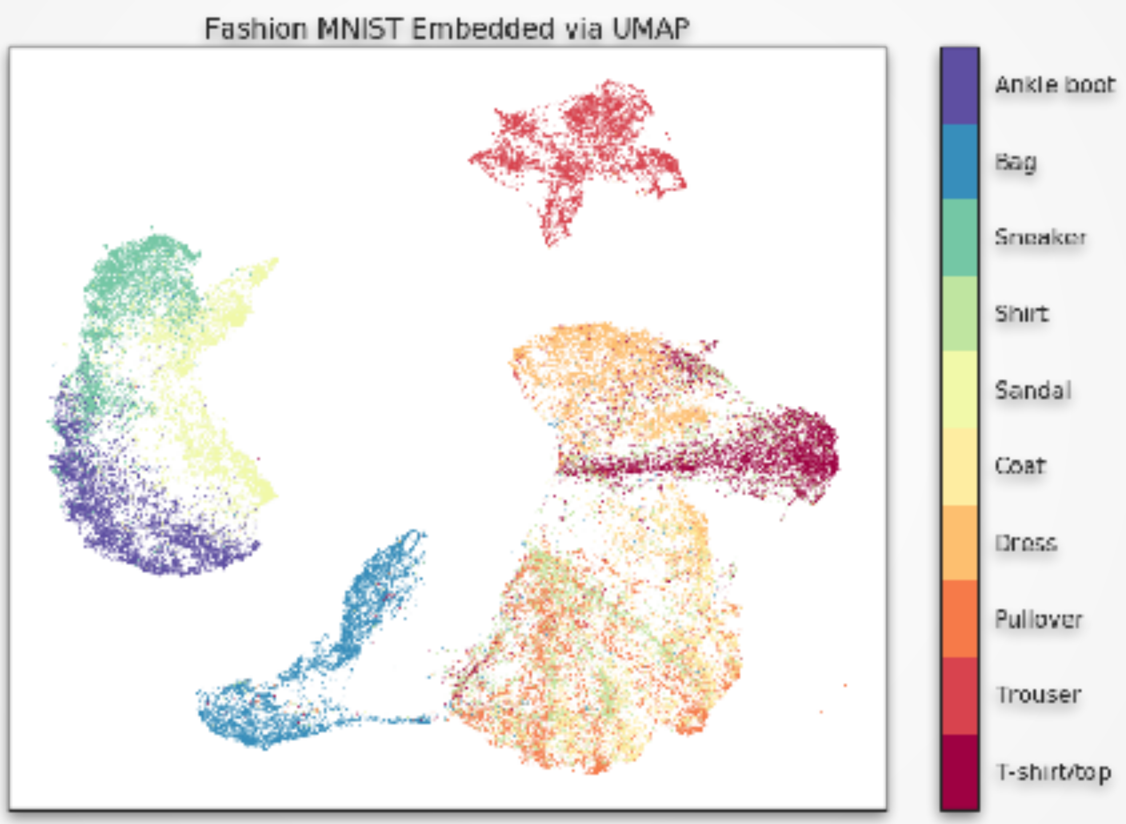
UMAP VizTech

- UMAP Uniform Manifold Approximation & Projection
- Preserves HD local structure as does *t*-SNE
- Often better at displaying global structure than *t*-SNE
- Constructs a topological representation of HD data
- For some LD representation, layout is optimized to minimize cross-entropy between HD & LD topological representations



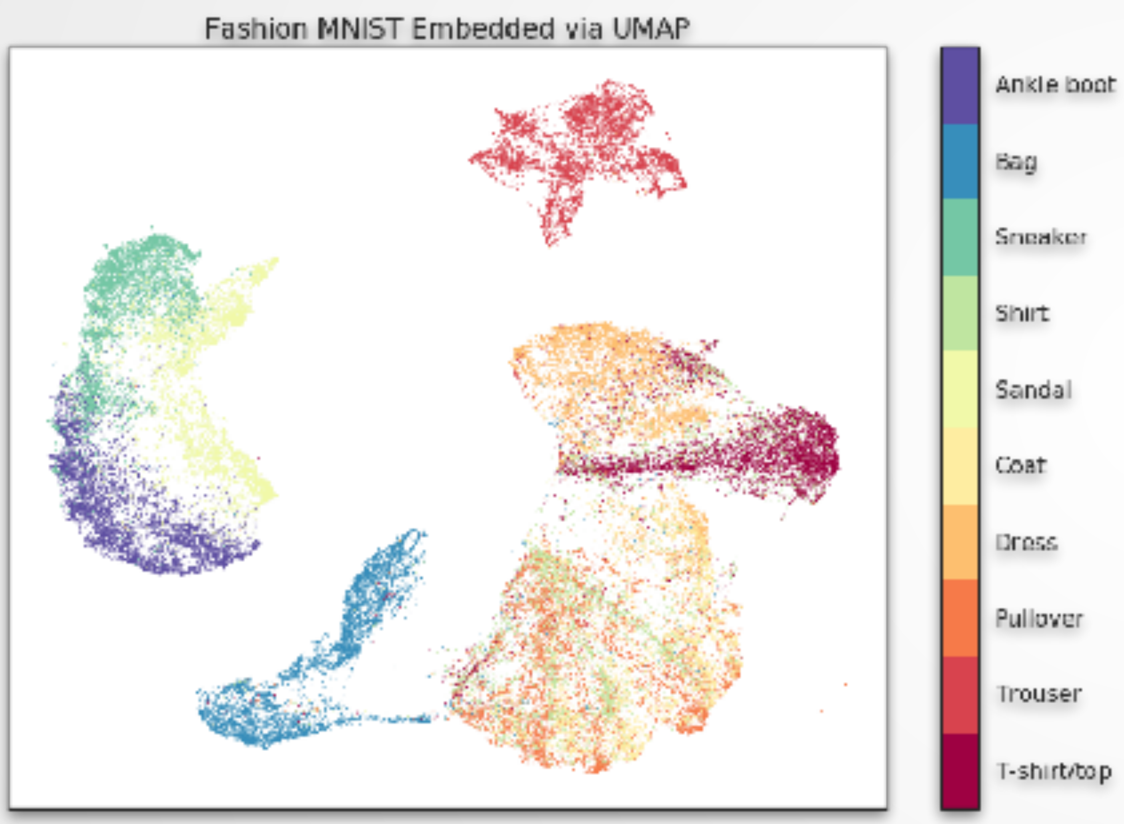
Source: UMAP documentation

UMAP VizTech



Source: UMAP documentation

UMAP VizTech



Source: UMAP documentation

t-SNE of Fashion MNIST. Source: Researchgate

UMAP intuition

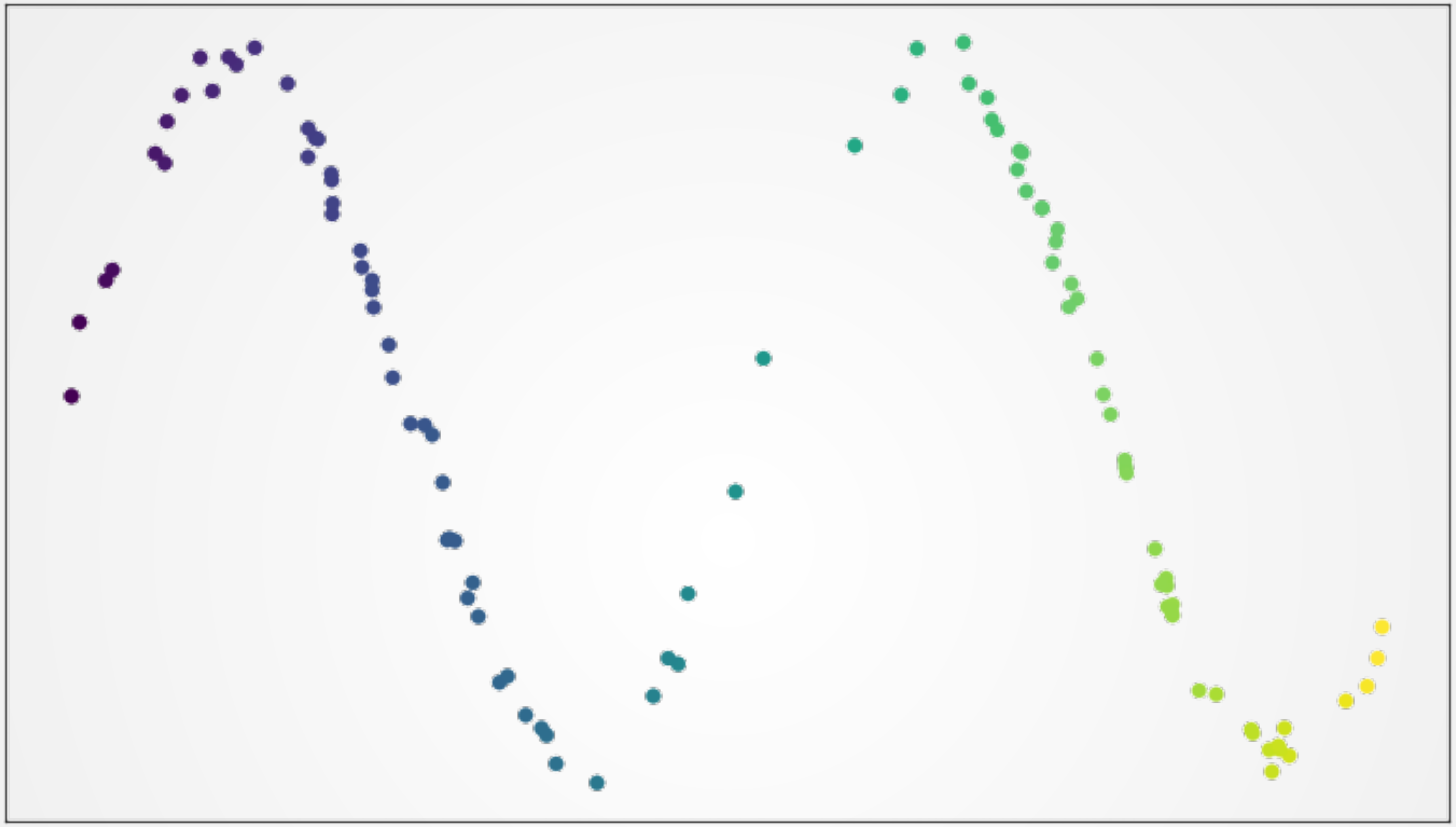


Image Source: UMAP documentation

Let us look at a nice sine curve.

Note that mesh points are not uniformly distributed

UMAP intuition

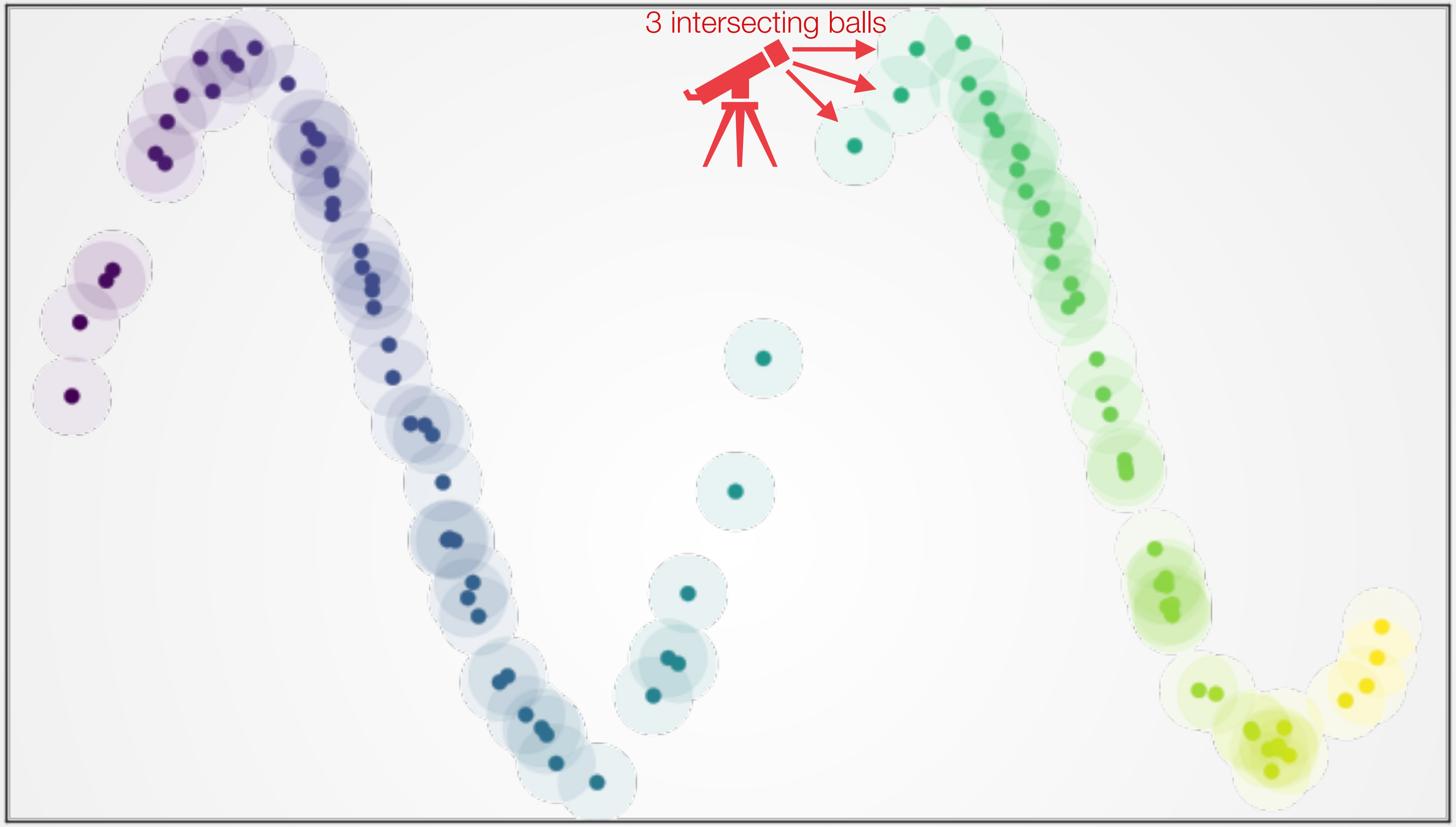


Image Source: UMAP documentation

- Generate an open cover
- Way to approximate open cover: balls of fixed radius around data

UMAP intuition

Look, how we convert balls into simplices!

UMAP intuition

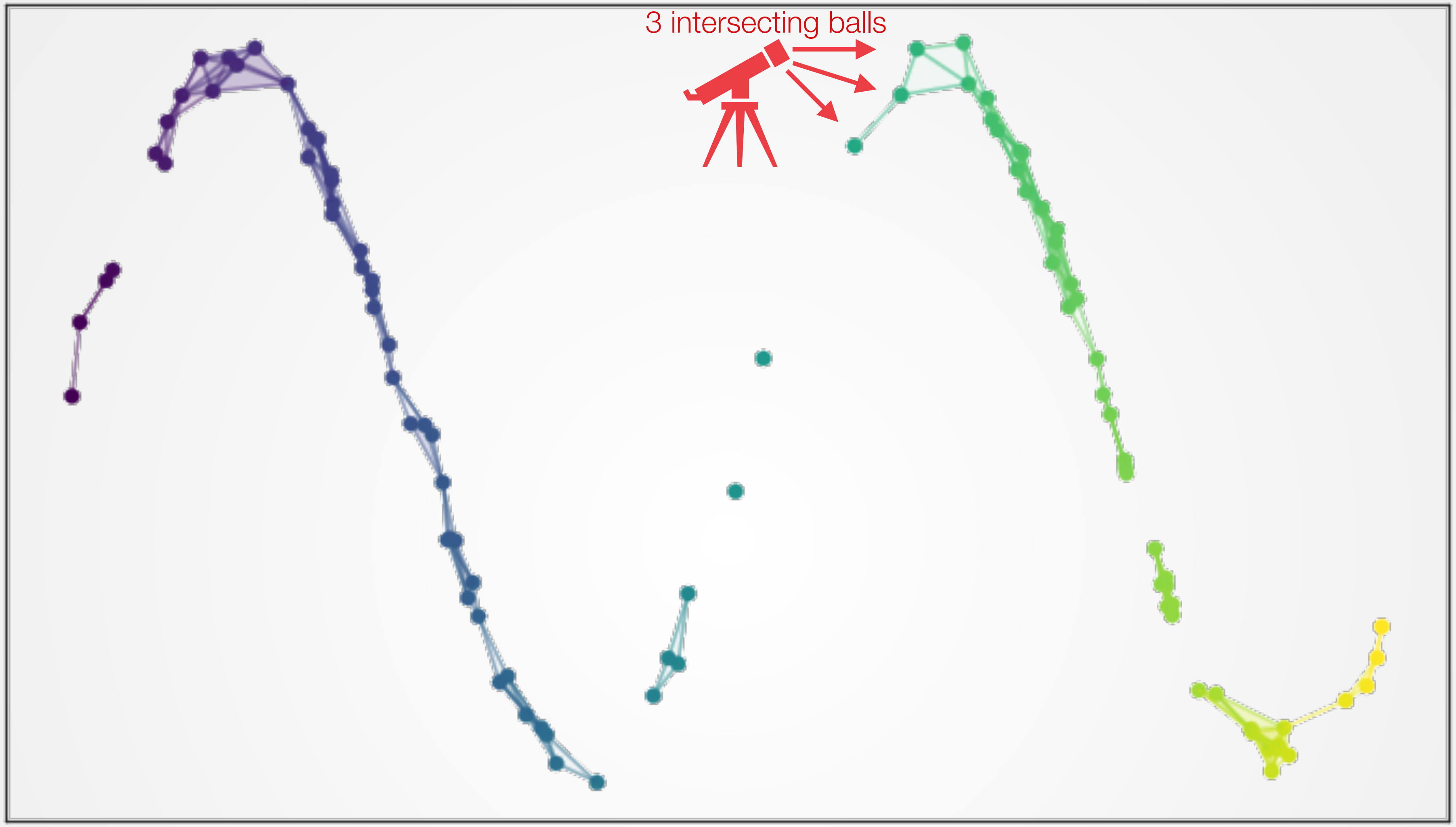


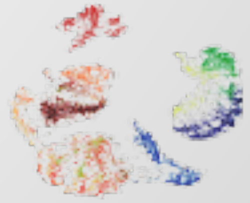
Image Source: UMAP documentation

Build simplicial complex of 0-, 1- and 2-simplices by connecting overlapping balls

UMAP intuition

Image Source: UMAP documentation

- Build simplicial complex of 0-, 1- and 2-simplices by connecting overlapping balls



UMAP intuition

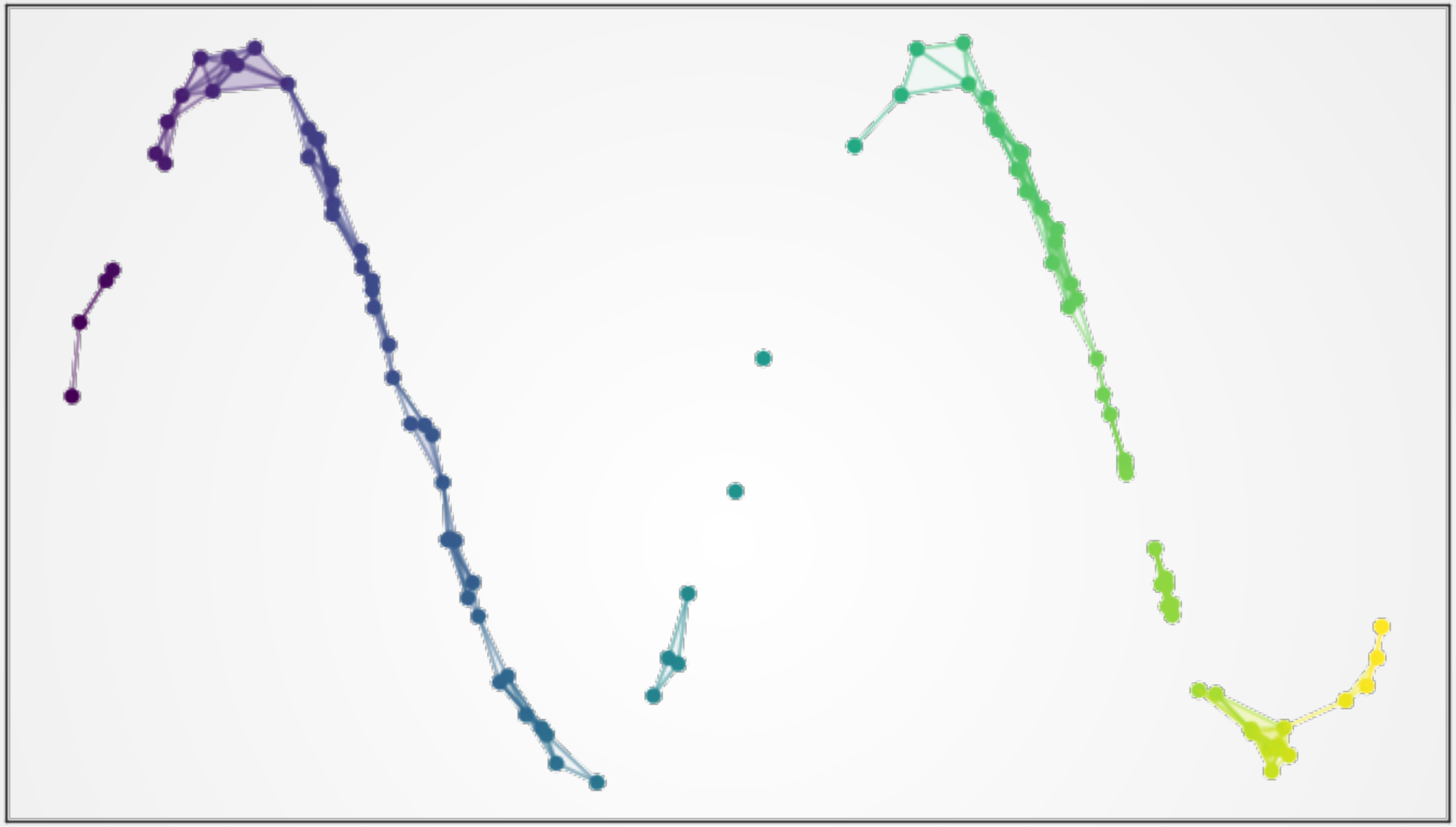


Image Source: UMAP documentation

- Build simplicial complex of 0-, 1- and 2-simplices
- Does a reasonable job of capturing fundamental topology

UMAP intuition

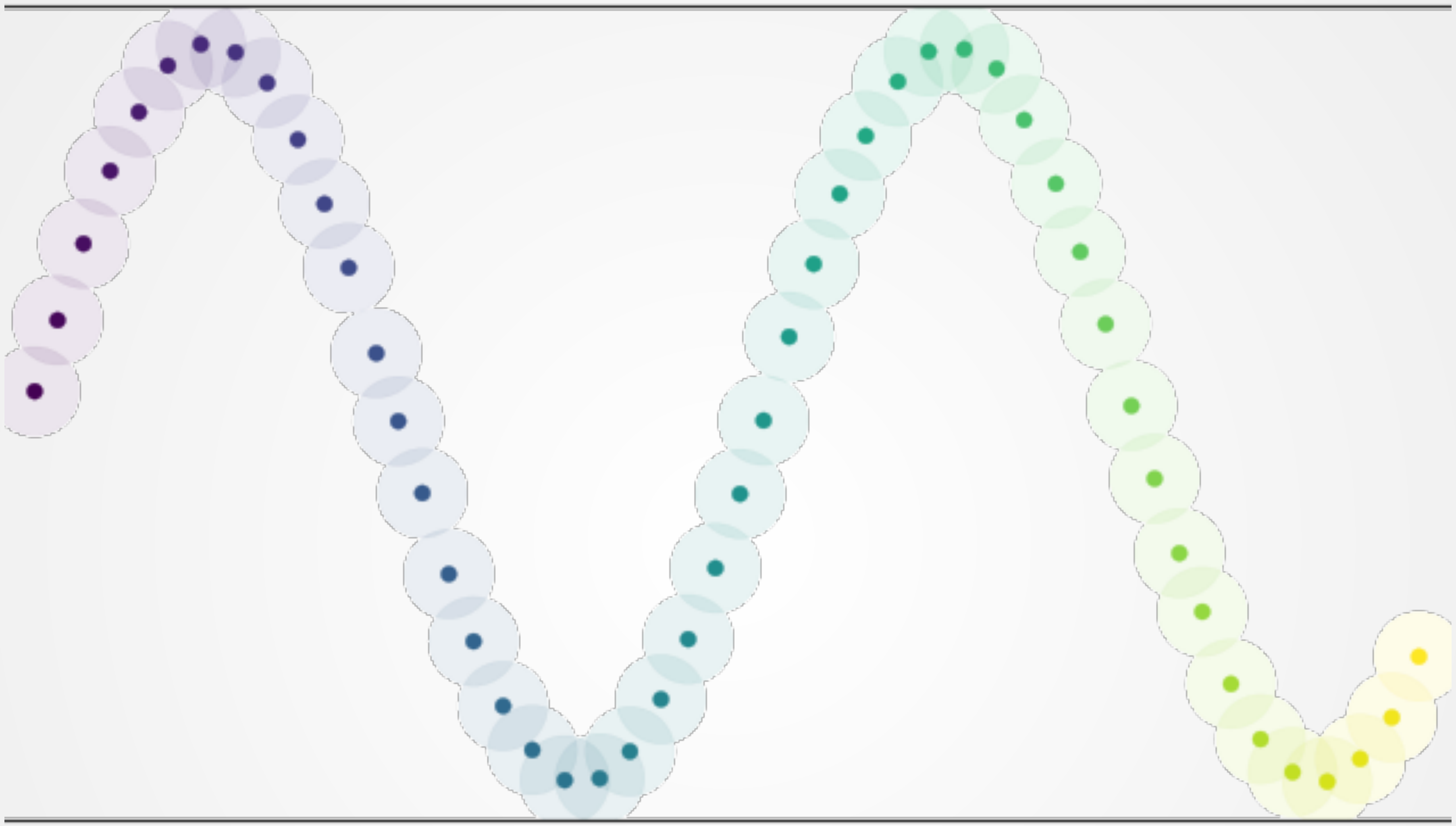
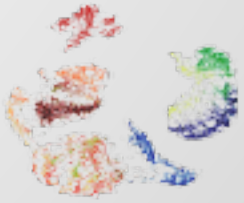


Image Source: UMAP documentation

- If data is uniformly distributed, open balls with the same radius work well
- But not in general!



UMAP intuition

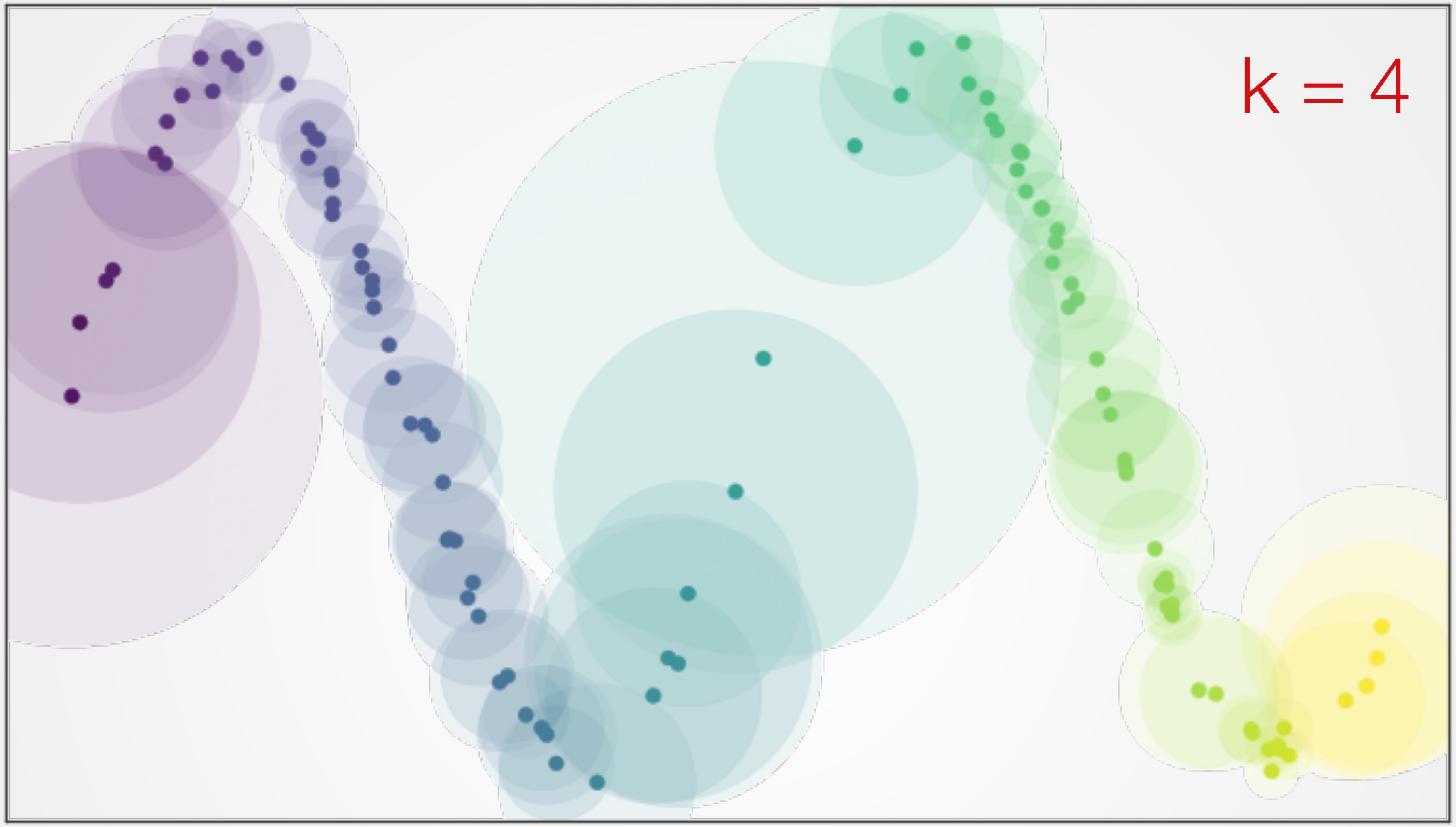


Image Source: UMAP documentation

- Let the „metric“ vary locally
- Still open balls with radius 1, but with varying metric
- Metric is chosen s.t. a ball of radius 1 touches *k* nearest neighbors

UMAP intuition

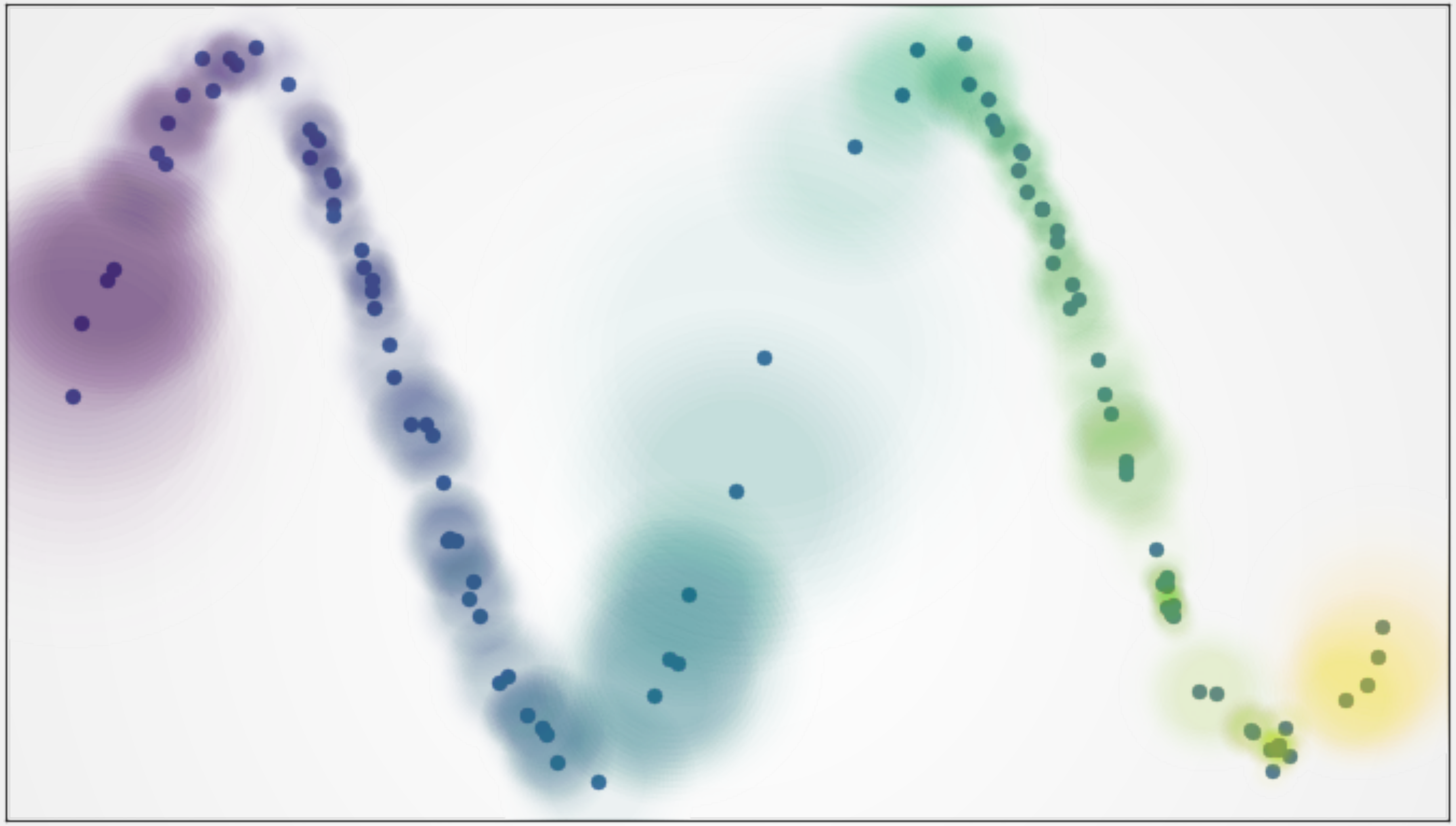
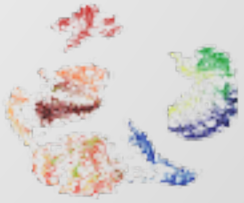


Image Source: UMAP documentation

- Refine to a fuzzy cover
- Interpret weights as probabilities of being connected



UMAP intuition

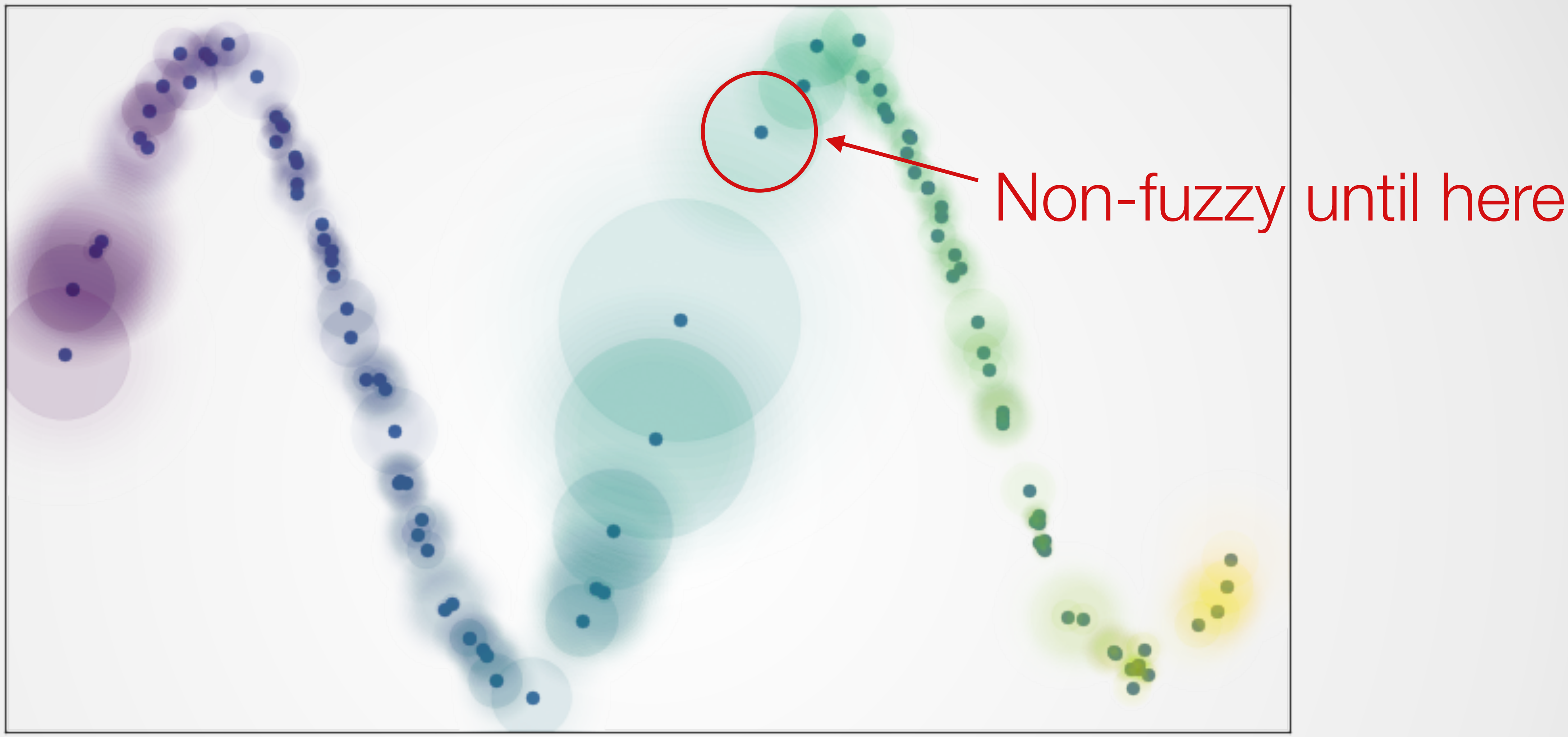


Image Source: UMAP documentation

- Leave no one (point) behind!
- To make sure no point is completely isolated, let fuzzy distance decay only after reaching the nearest neighbor

UMAP intuition

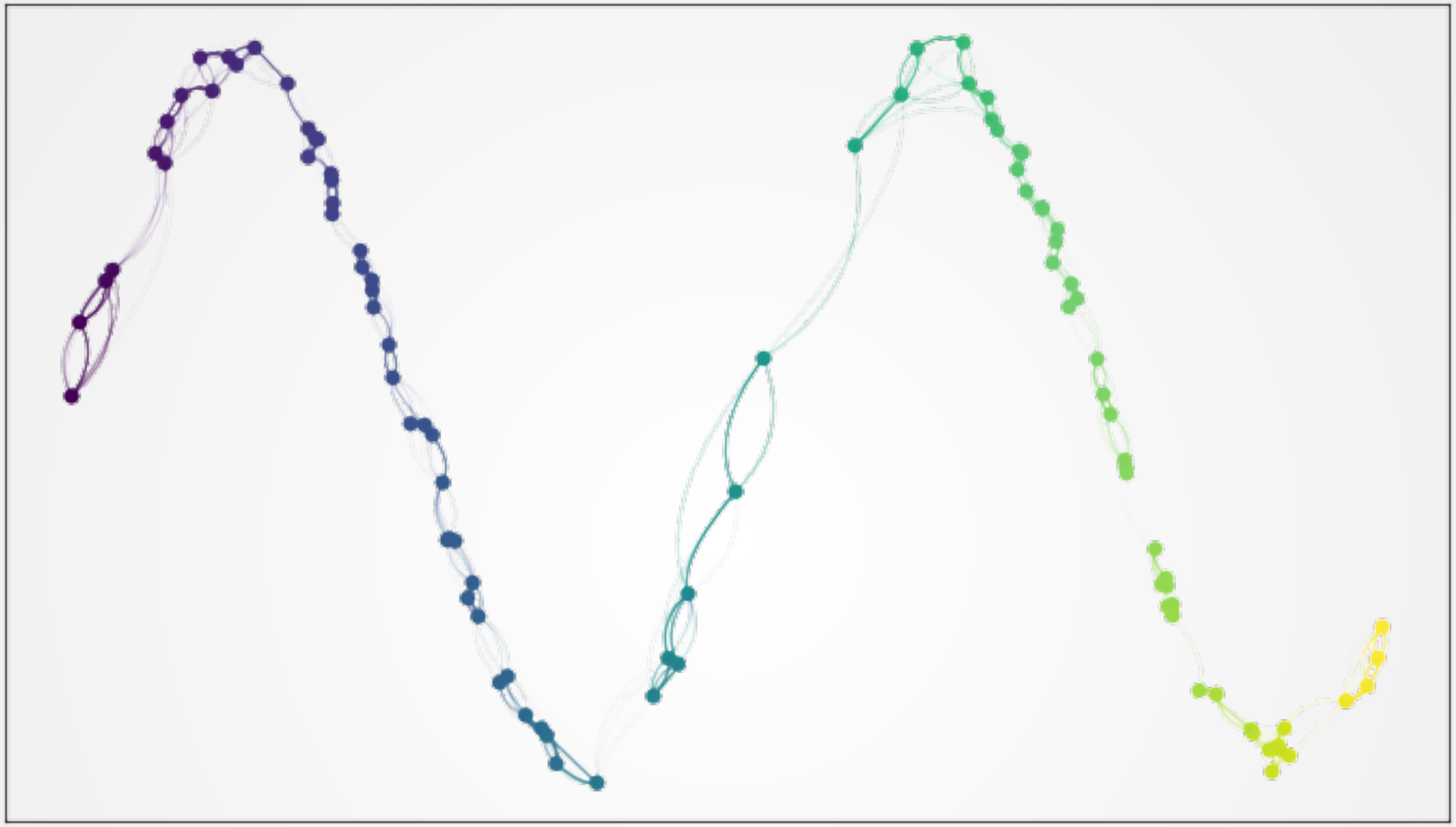


Image Source: UMAP documentation

- Problem: local metrics are incompatible. Edges can have different weights depending on metrics of endpoints
- Merge disagreeing two edges a and b

UMAP intuition

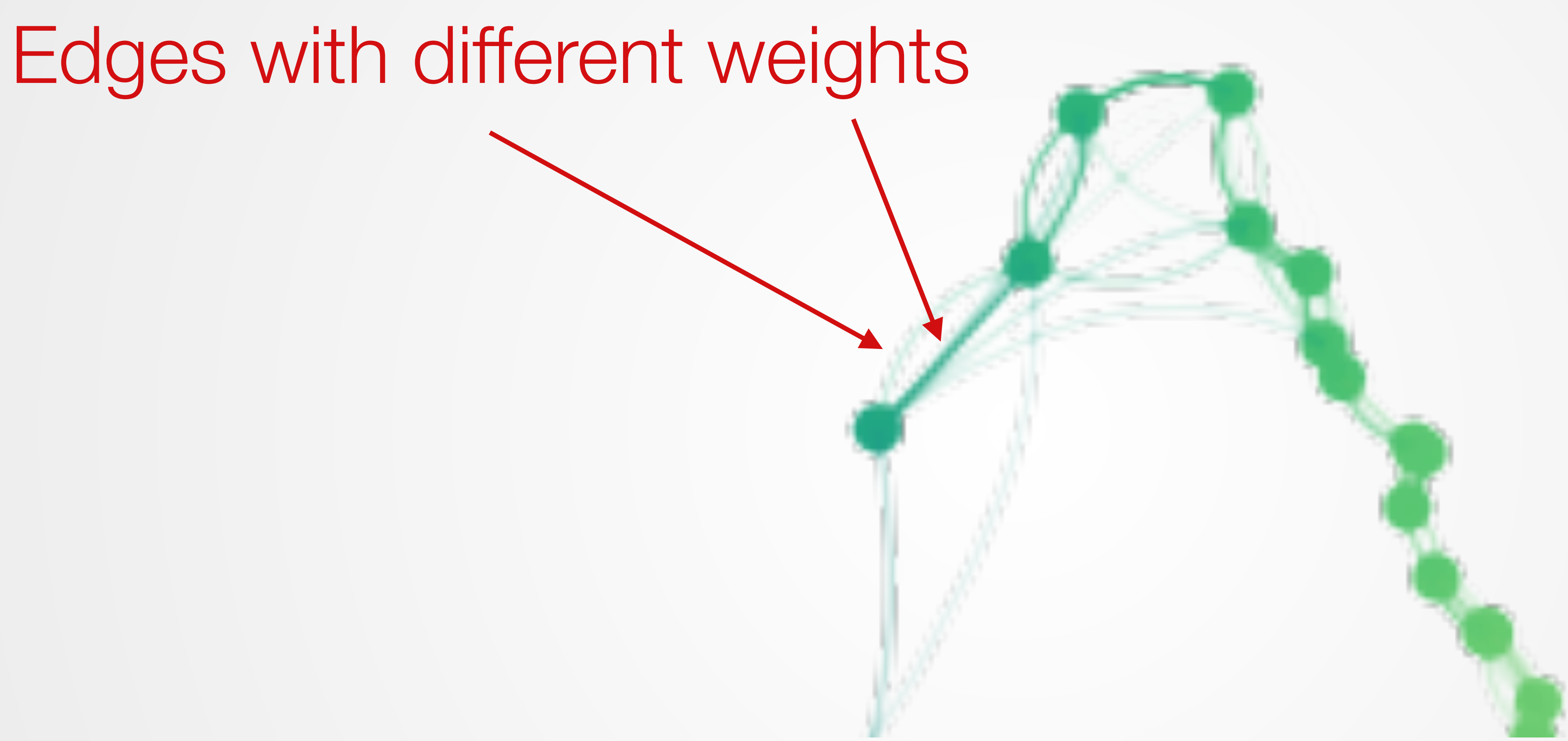
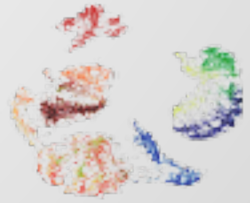


Image Source: UMAP documentation

- Problem: local metrics are incompatible. Edges can have different weights depending on metrics of endpoints
- Merge disagreeing two edges a and b



UMAP intuition

Extreme case: there can even be only one edge

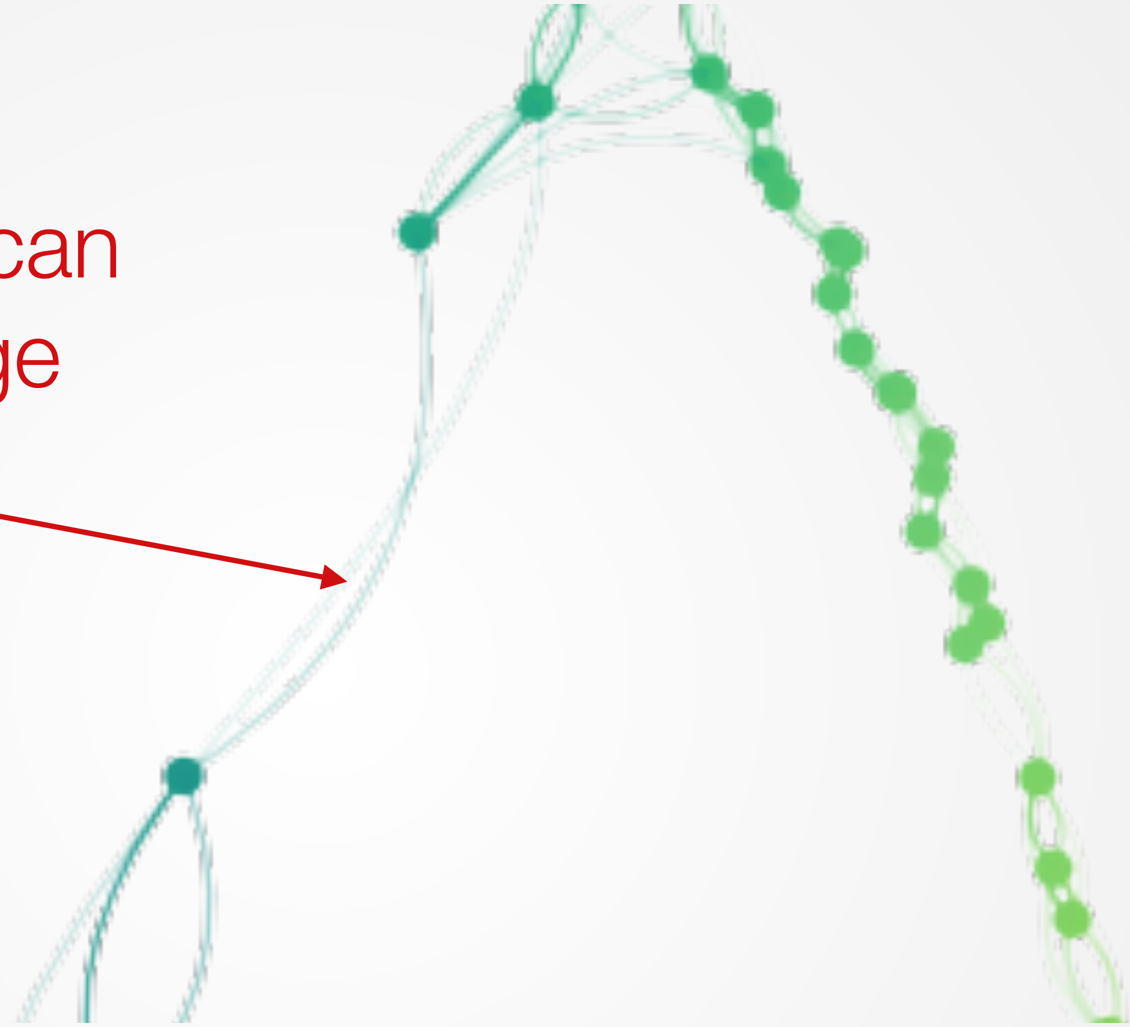


Image Source: UMAP documentation

- Problem: local metrics are incompatible. Edges can have different weights depending on metrics of endpoints
- Merge disagreeing two edges a and b

UMAP intuition

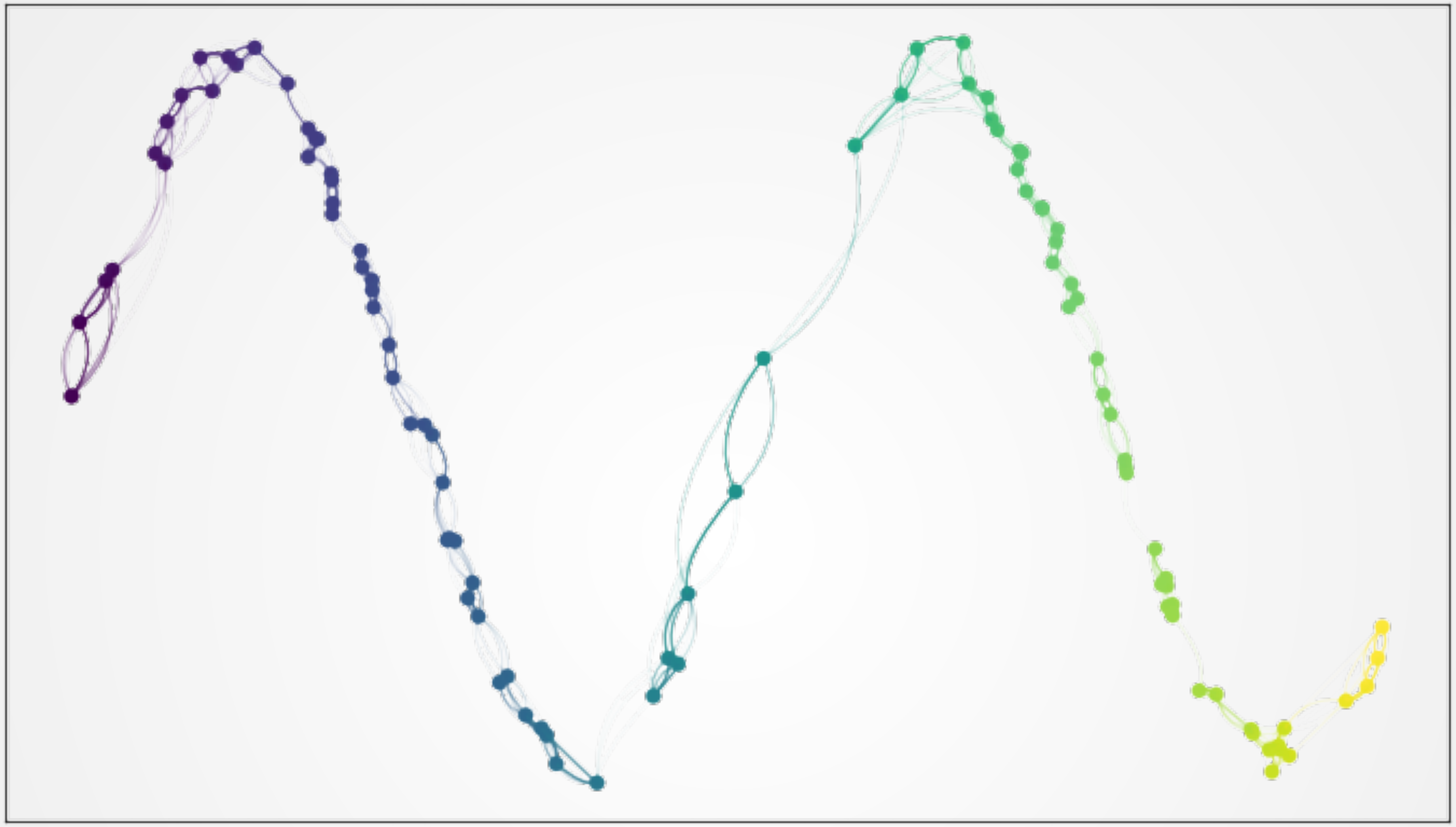


Image Source: UMAP documentation

- Merge disagreeing two edges a and b
- Combined weight: $a + b - ab$
- Interpretation: probability that at least one of the edges exists

UMAP intuition

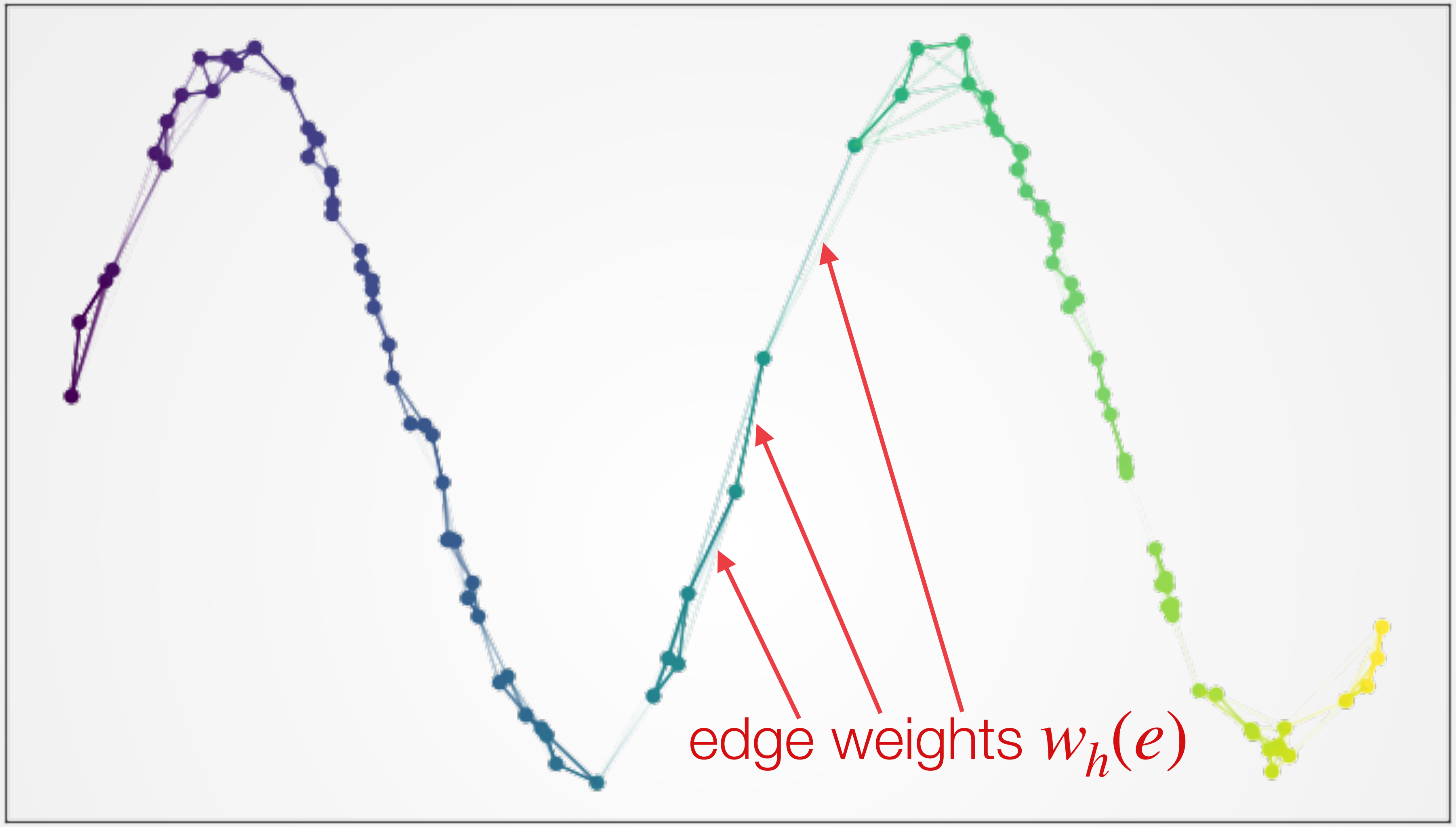


Image Source: UMAP documentation

- Graph with combined edge weights

UMAP intuition

- Main step: find representation in low dim. that is „close“ to high dim.
- How to measure closeness?
  - Compare vectors of 1-simplices = probabilities = edge weights between 0-simplices (data points)
  - These are Bernoulli random variables, edge either exists or not
  - Cross-entropy is an appropriate measure
- Minimize cross-entropy leads to maximum „closeness"

UMAP intuition

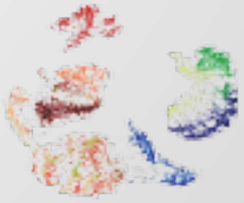
- Cross-entropy:

$$\sum_{e \in E} w_h(e) * \log \left\{ \frac{w_h(e)}{w_l(e)} \right\} + \{1 - w_h(e)\} * \log \left\{ \frac{1 - w_h(e)}{1 - w_l(e)} \right\}$$

E: set of edges

$w_h(e)$: edge weight in high dimension

$w_l(e)$: edge weight in low dimension



UMAP intuition

- Cross-entropy:

$$\sum_{e \in E} \boxed{w_h(e) * \log\left\{\frac{w_h(e)}{w_l(e)}\right\}} + \{1 - w_h(e)\} * \log\left\{\frac{1 - w_h(e)}{1 - w_l(e)}\right\}$$

Attractive force: Minimized when $w_l(e)$ is as large as possible.

Large weights = small distances

UMAP intuition

- Cross-entropy:

$$\sum_{e \in E} w_h(e) * \log \left\{ \frac{w_h(e)}{w_l(e)} \right\} + \boxed{\{1 - w_h(e)\} * \log \left\{ \frac{1 - w_h(e)}{1 - w_l(e)} \right\}}$$

Repulsive force: Minimized when $w_l(e)$ is as small as possible

Remember: Small weights = large distances

Attractive force: Minimized when $w_h(e)$ is as large as possible

Remember: Large weights = small distances

UMAP intuition

- Cross-entropy:

$$\sum_{e \in E} \boxed{w_h(e) * \log\left\{\frac{w_h(e)}{w_l(e)}\right\}} + \boxed{\{1 - w_h(e)\} * \log\left\{\frac{1 - w_h(e)}{1 - w_l(e)}\right\}}$$

Balance of pull and push lets low dimensional representation settle into accurate representation of high dimensional topology

Repulsive force: Minimized when $w_h(e)$ is as small as possible

Remember: Small weights = large distances

Attractive force: Minimized when $w_h(e)$ is as large as possible

Remember: Large weights = small distances

UMAP Hyperparameters

- Number of neighbors
  - How many neighbors are considered for open balls?
  - Controls local vs. global
- Minimal distance
  - How tight can points in lower dimensional space be?
- Number components
  - How many dimensions should lower dimensional space have?
- Metric
  - What metric is used?
  - e.g. Euclidean, Manhattan, Mahalanobis, etc.

Image data: CryptoPunks

- 10000 images, each 24×24 pixels, RGB colour space
- Image data are high-dimensional
- Flatten the data by converting each image to an array
- Remove noninformative pixels (pixels that are black in every image)
- Hyperparameter settings in the following examples
  - n_neighbors = 10
  - min_dist = 0.1
  - n_components = 2
  - metric = euclidean

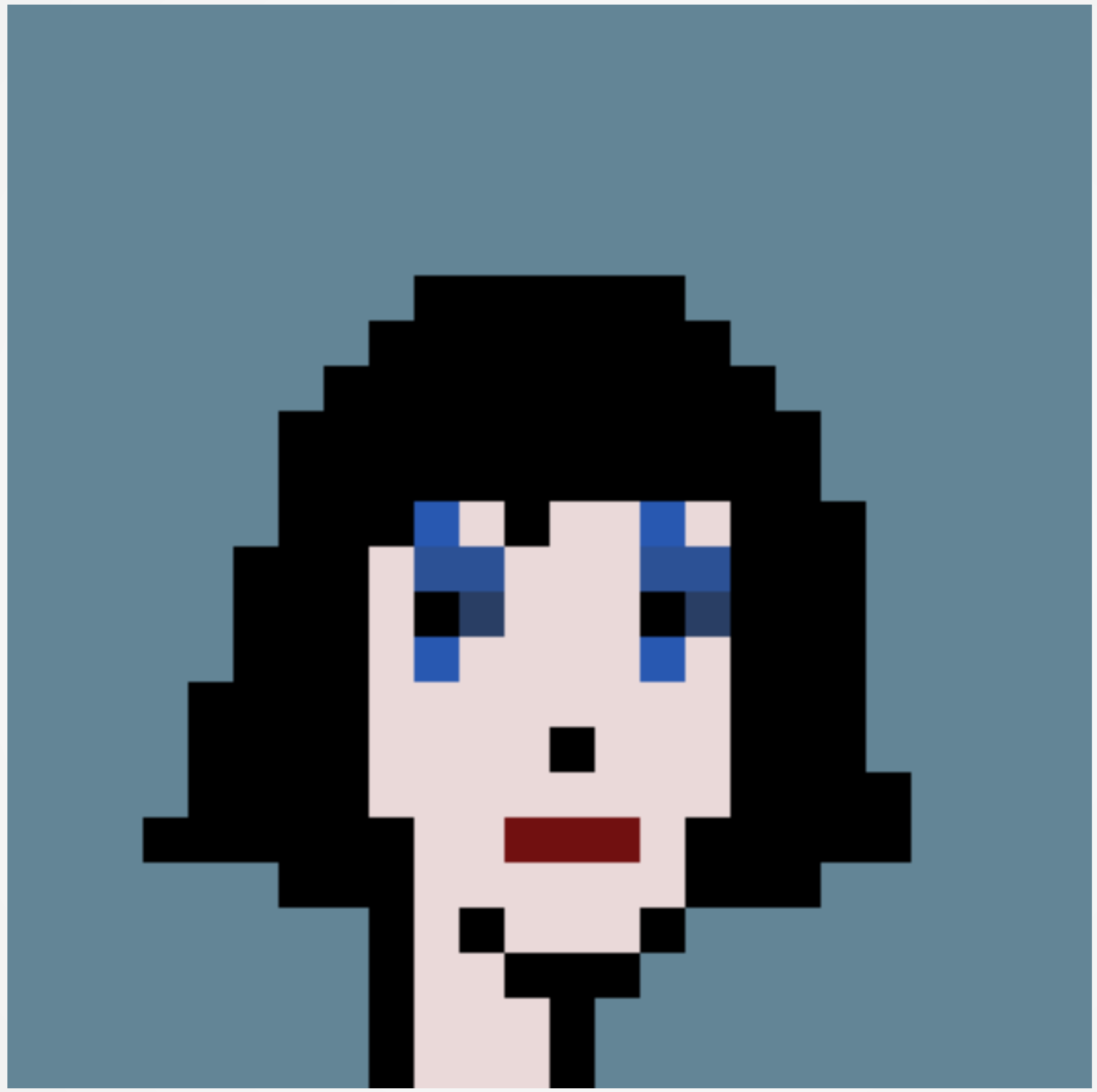
Image data: CryptoPunks

3777
Check on
interactive plot

Click here for interactive plot

UMAP

Image data: CryptoPunks



an isolated Punk?!

Click here for interactive plot

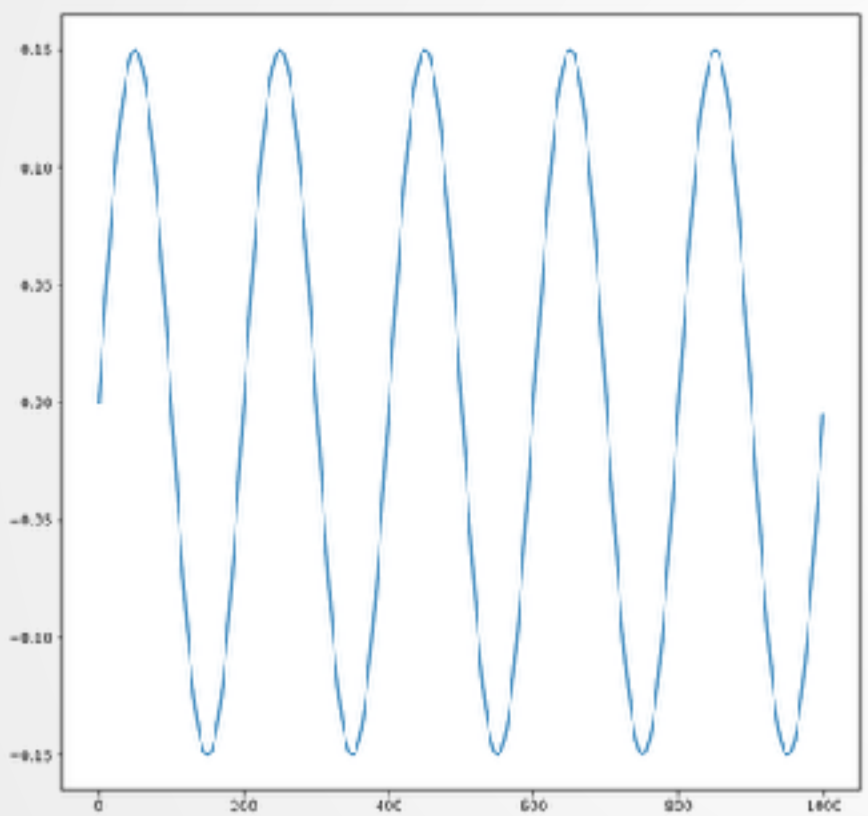
UMAP

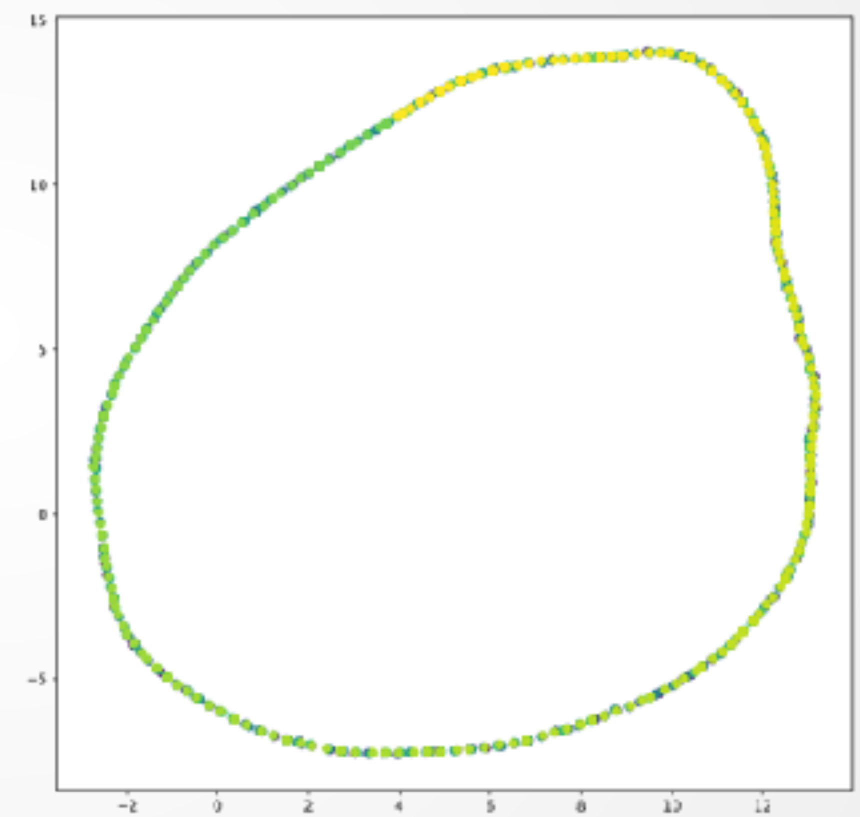
Illustrative examples

- Challenge are time series!
- Univariate here, multivariate time series later
- Time series transformed in a sequence of rolling windows
- Window at time t: data point x_t
- Window width $w \succ \mathbb{R}^w$ dimensional data points
- Default hyperparameter settings in the following examples
  - n_neighbors = 15
  - min_dist = 0.1
  - n_components = 2
  - metric = euclidean

Illustrative examples with simulated data

- Example 1: Simple Sine curve, period = 200, n = 1000
- Window = lag size: *d=5*

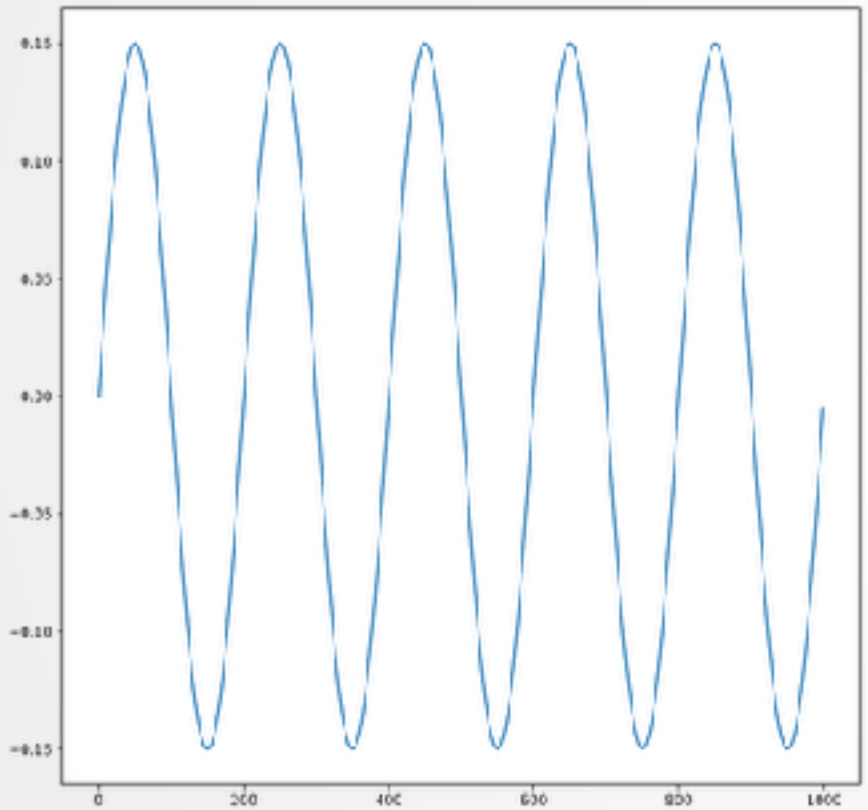


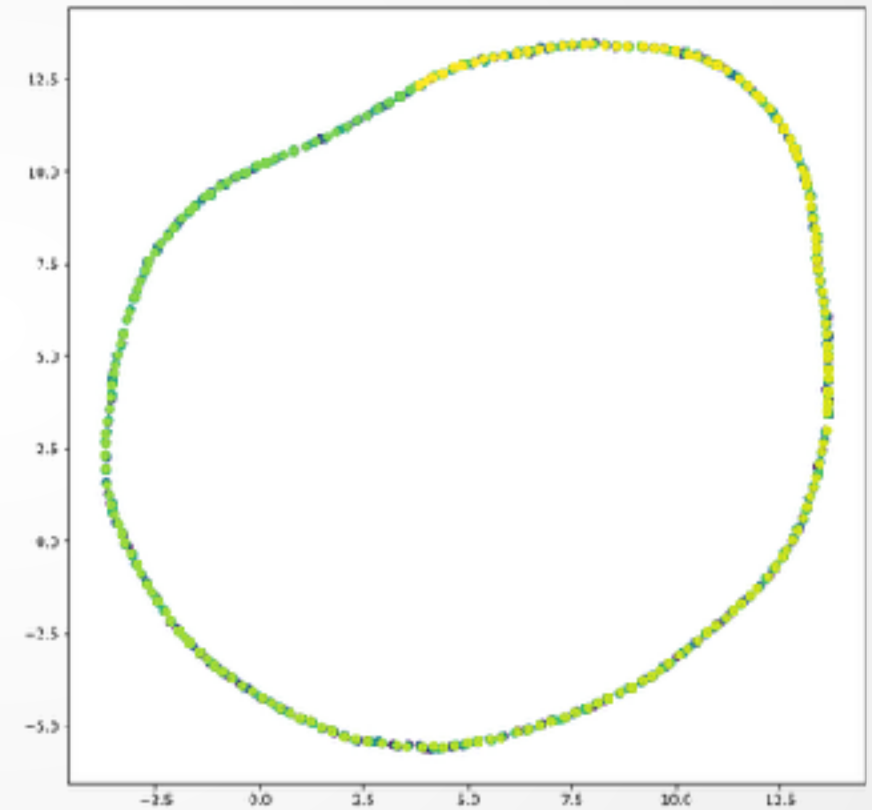


- UMAP recovers periodic structure well
- Interpretation: „data goes in circles“

Illustrative examples with simulated data

- Example 1: Simple Sine curve, period = 200, n = 1000
- Window = lag size: *d=30*

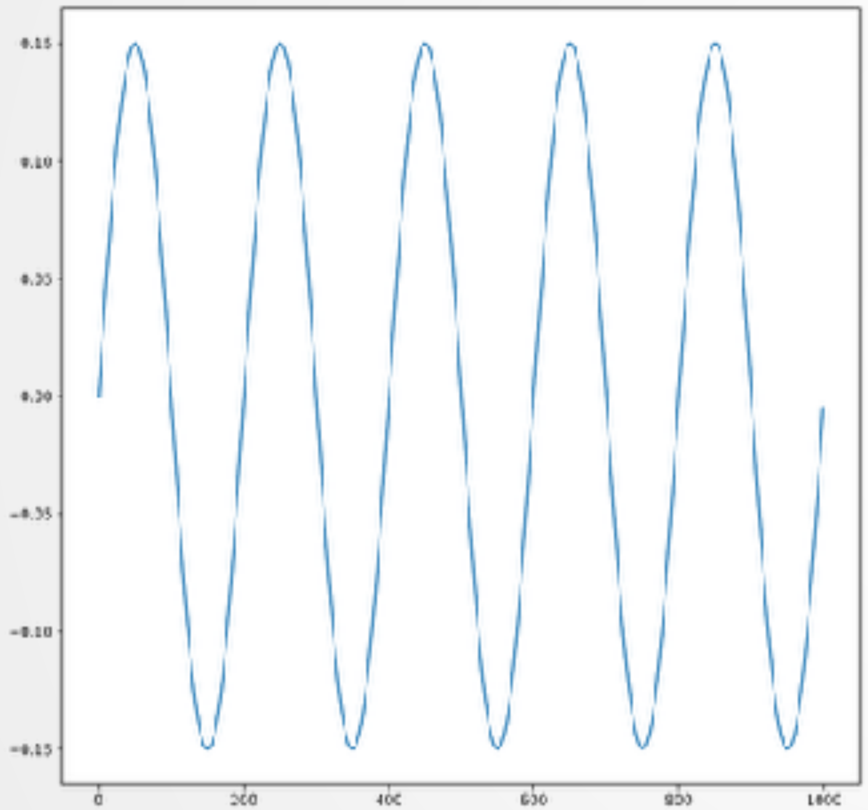


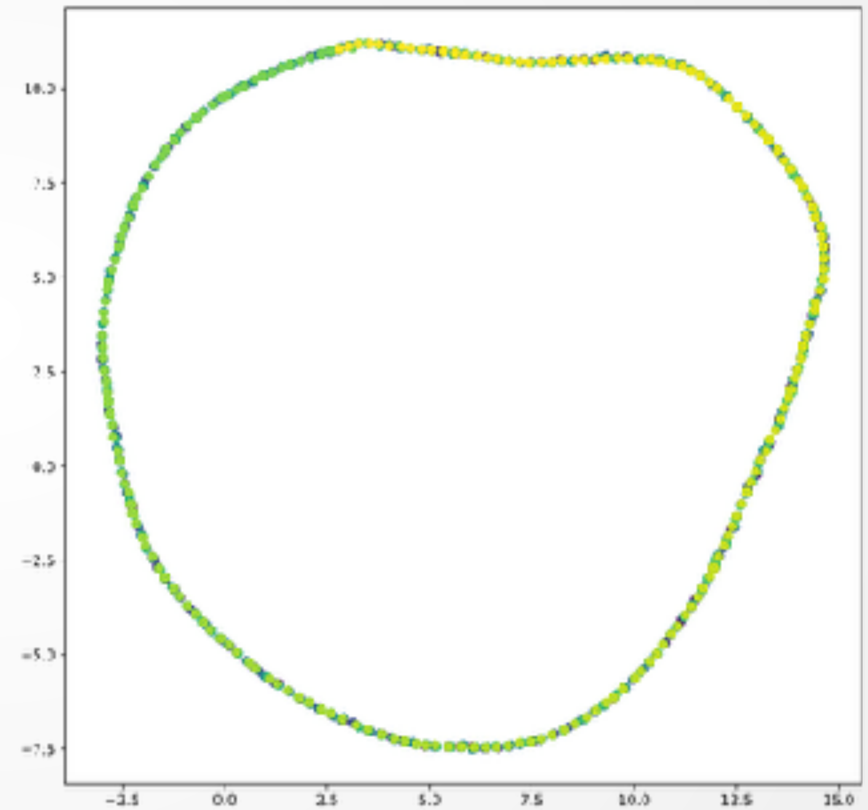


- UMAP recovers periodic structure well
- Interpretation: „data goes in circles“

Illustrative examples with simulated data

- Example 1: Simple Sine curve, period = 200, n = 1000
- Window = lag size: *d=50*

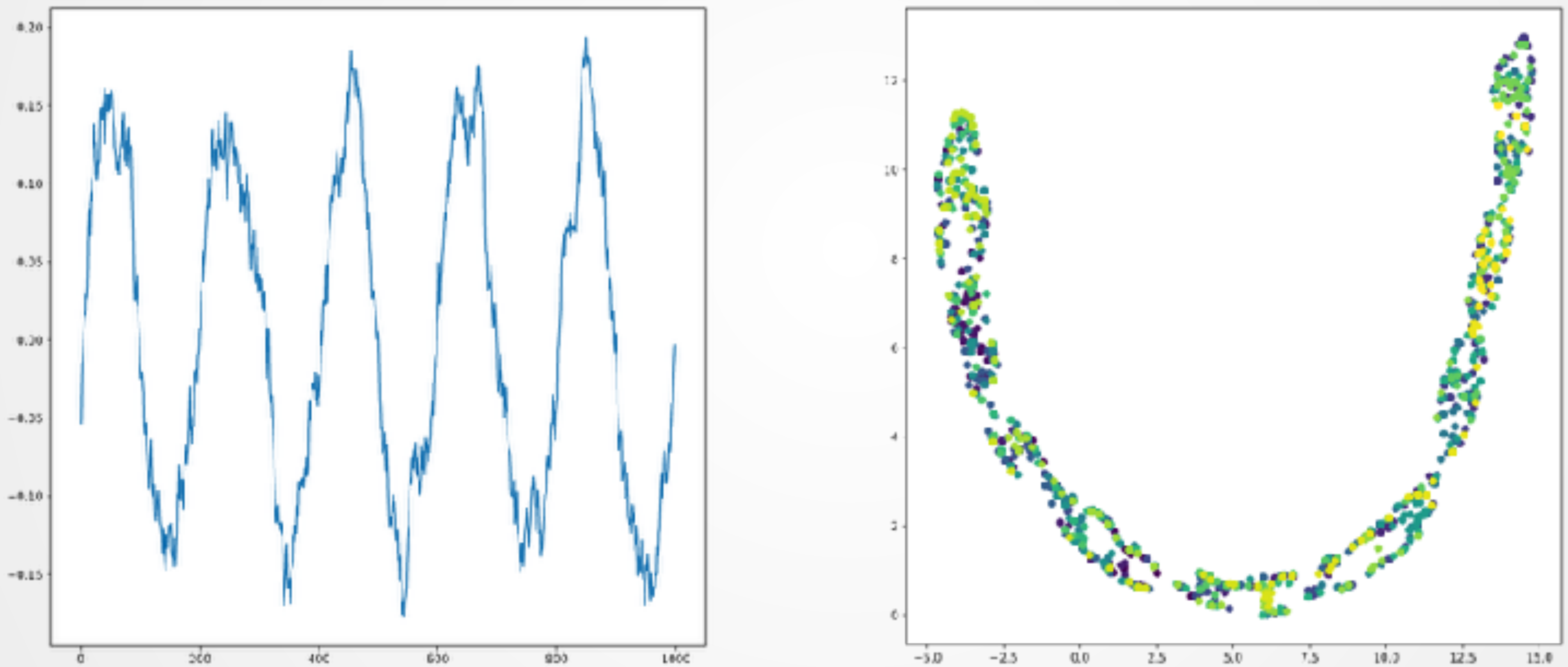




- UMAP recovers periodic structure well
- Interpretation: „data goes in circles“

Illustrative examples with simulated data

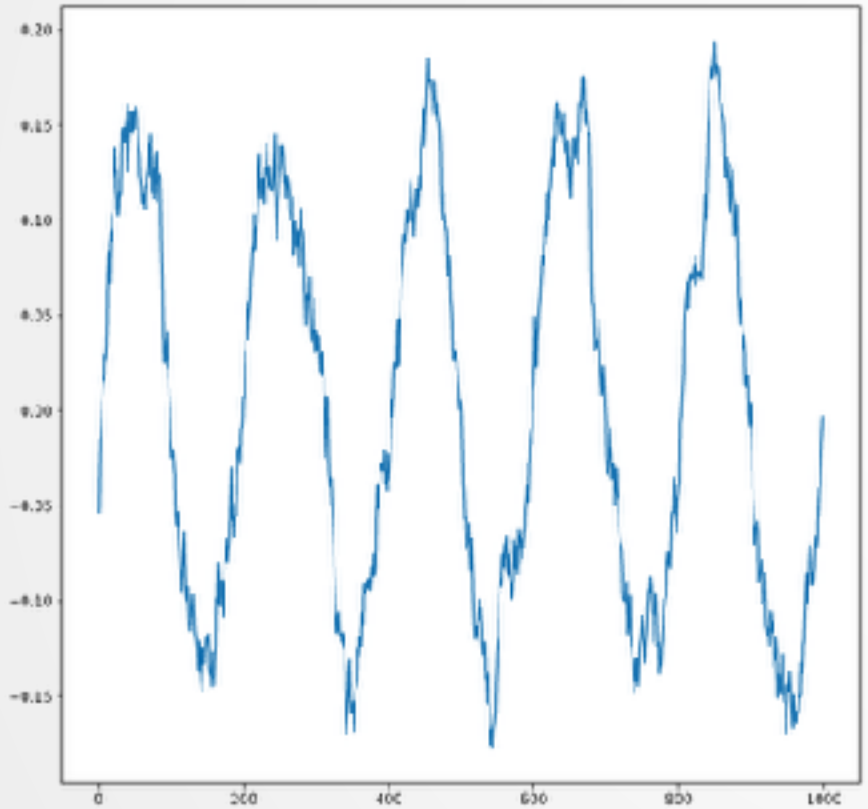
- Example 2: Sine curve from example 1 with added AR(1) process
- Autoregressive parameter: 0.9, Error sd: 0.1
- Window = lag size: *d=5*

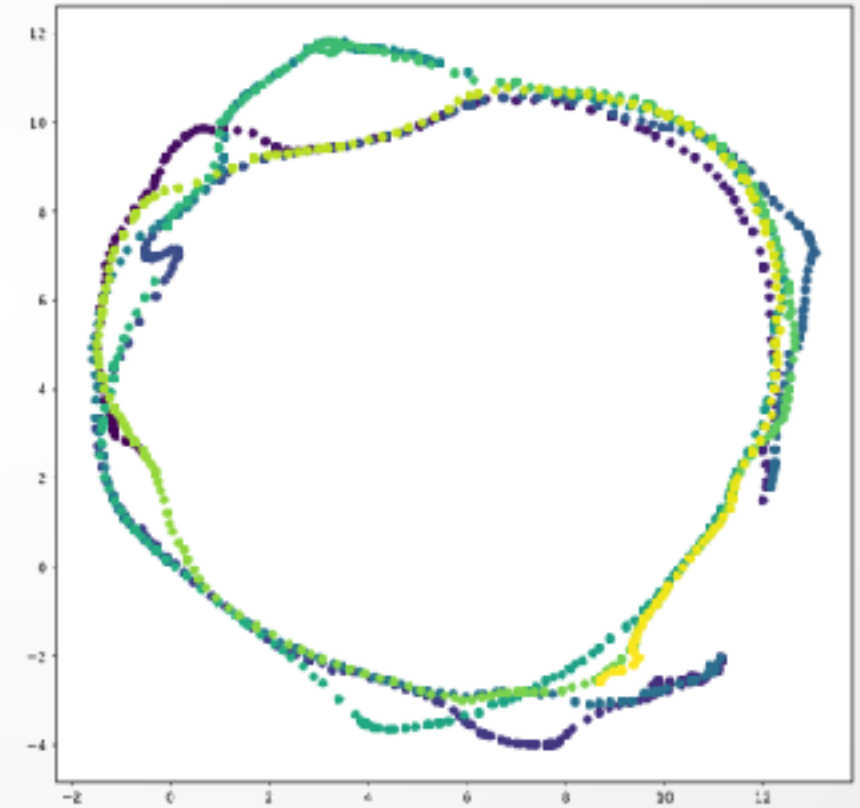


- UMAP still recovers periodicity
- Even though the structure of the data is more irregular

Illustrative examples with simulated data

- Example 2: Sine curve from example 1 with added AR(1) process
- Autoregressive parameter: 0.9, Error sd: 0.1
- Window = lag size: *d=30*

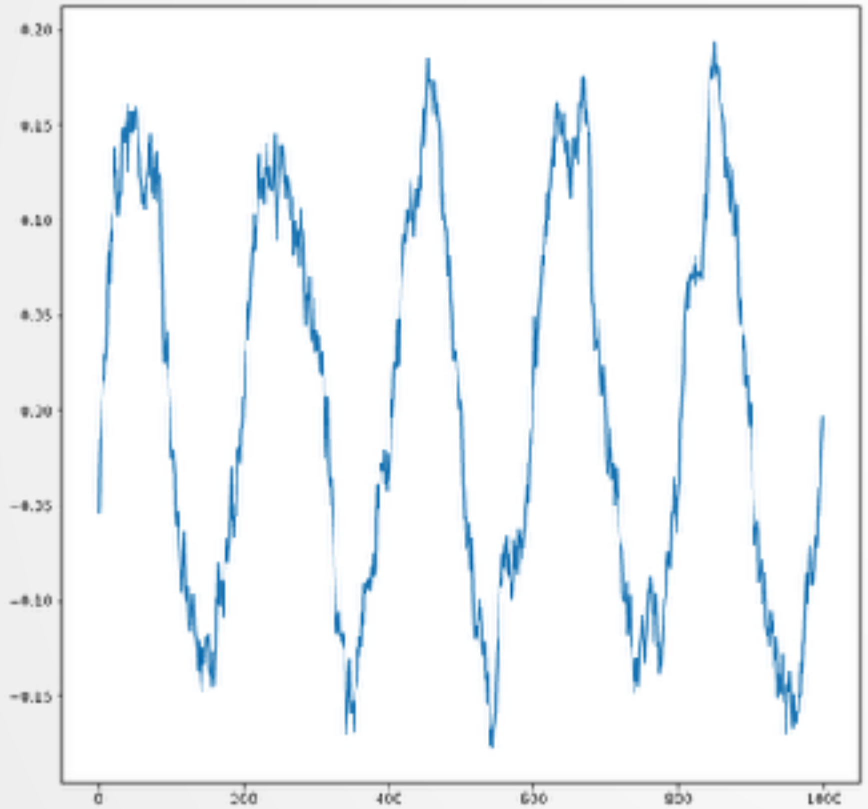


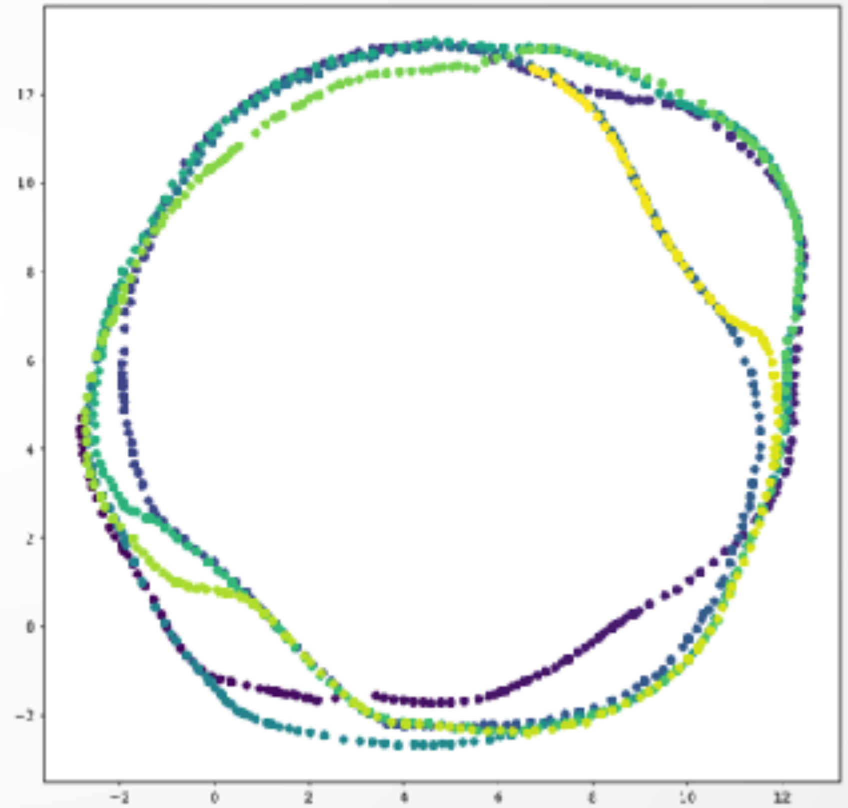


- UMAP still recovers periodicity
- Even though the structure of the data is more irregular

Illustrative examples with simulated data

- Example 2: Sine curve from example 1 with added AR(1) process
- Autoregressive parameter: 0.9, Error sd: 0.1
- Window = lag size: *d=50*

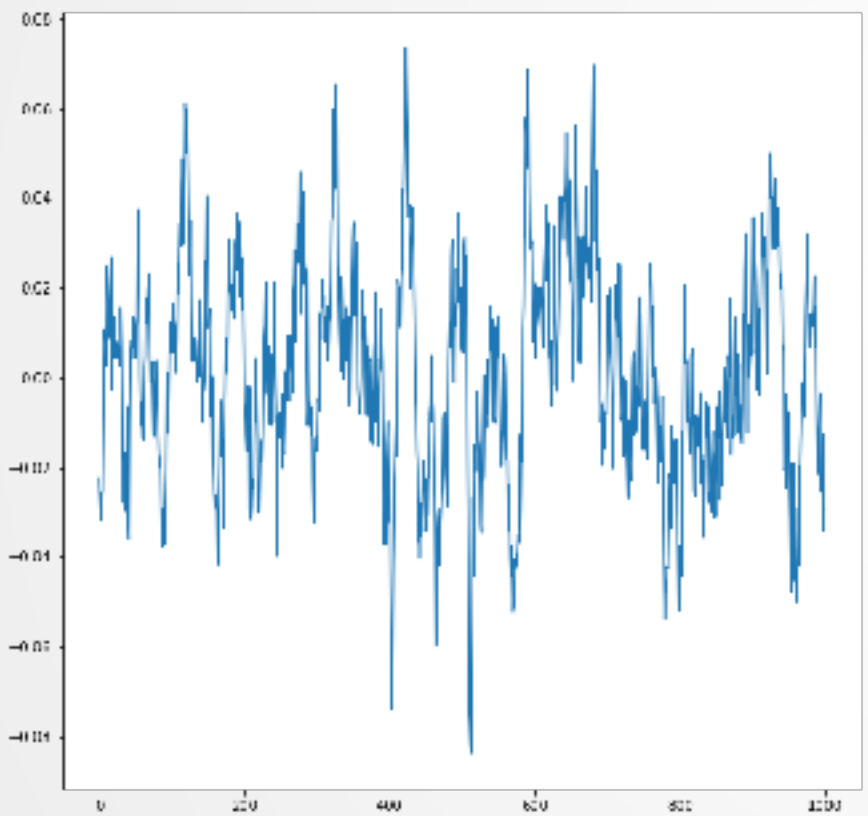


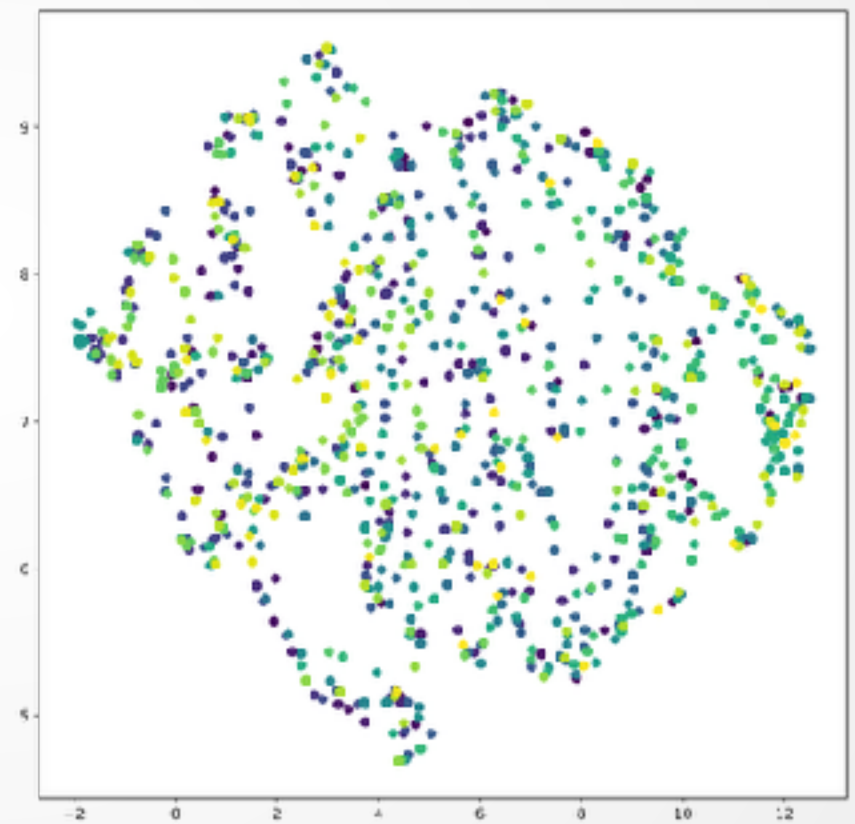


- UMAP still recovers periodicity
- Even though the structure of the data is more irregular

Illustrative examples with simulated data

- Example 3: Only AR(1) process
- Autoregressive parameter: 0.9 Error sd: 0.1
- Window = lag size: *d=5*

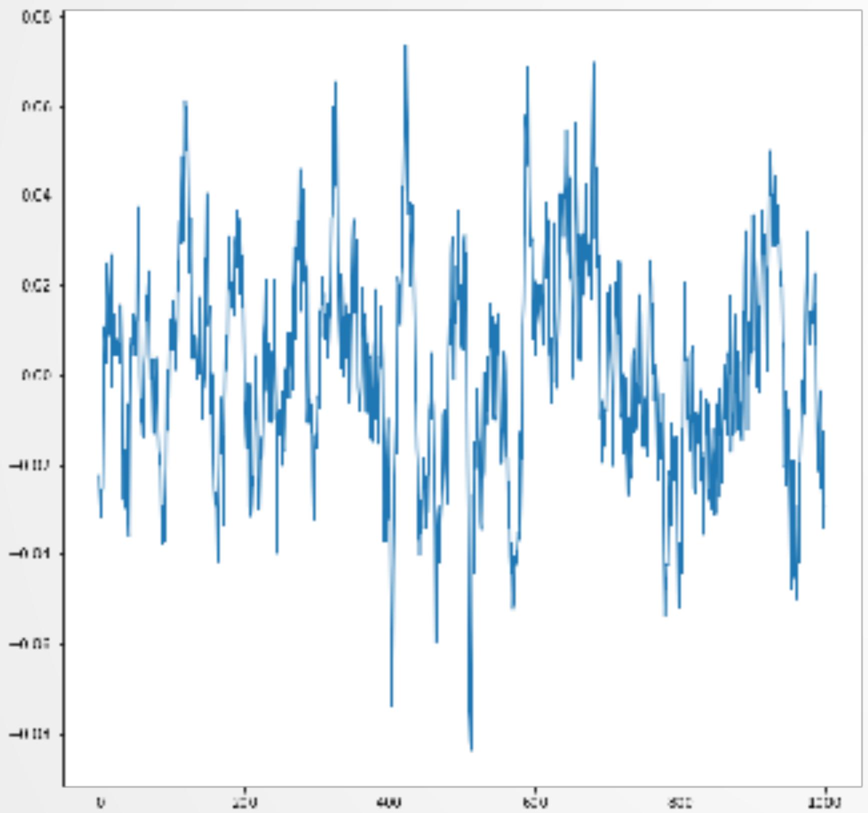


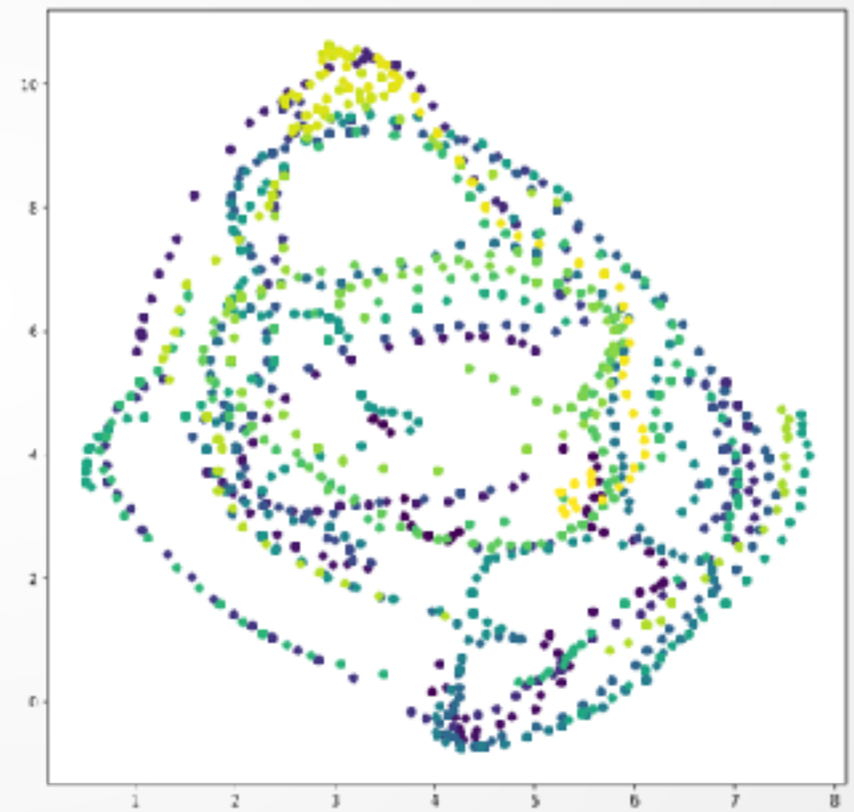


- UMAP recovers local structure for sufficient window size

Illustrative examples with simulated data

- Example 3: Only AR(1) process
- Autoregressive parameter: 0.9 Error sd: 0.1
- Window = lag size: *d=30*

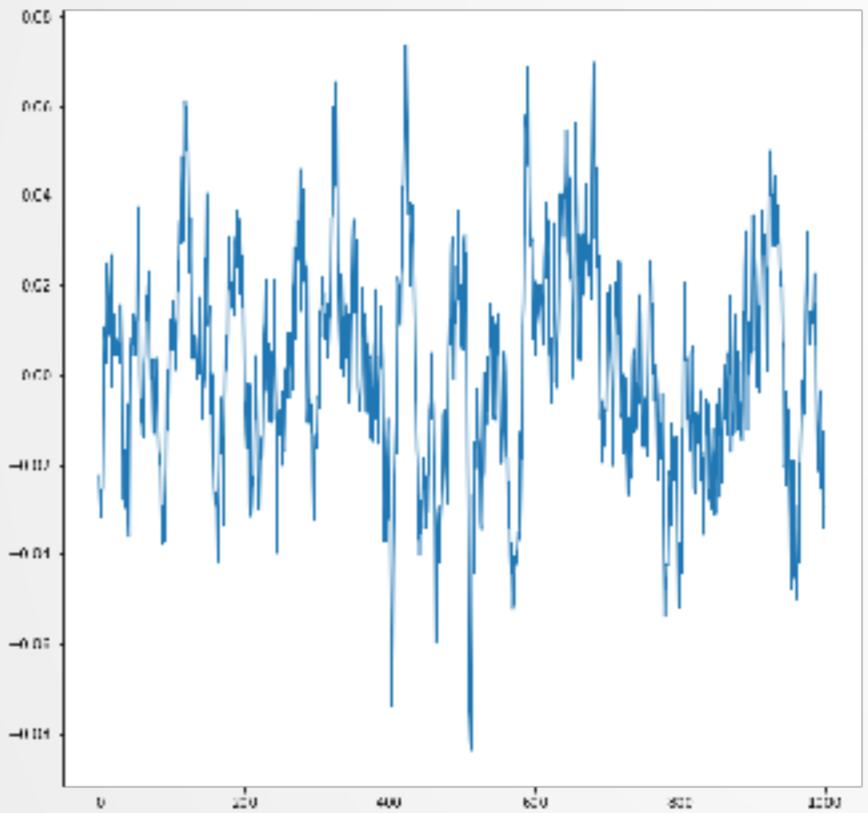


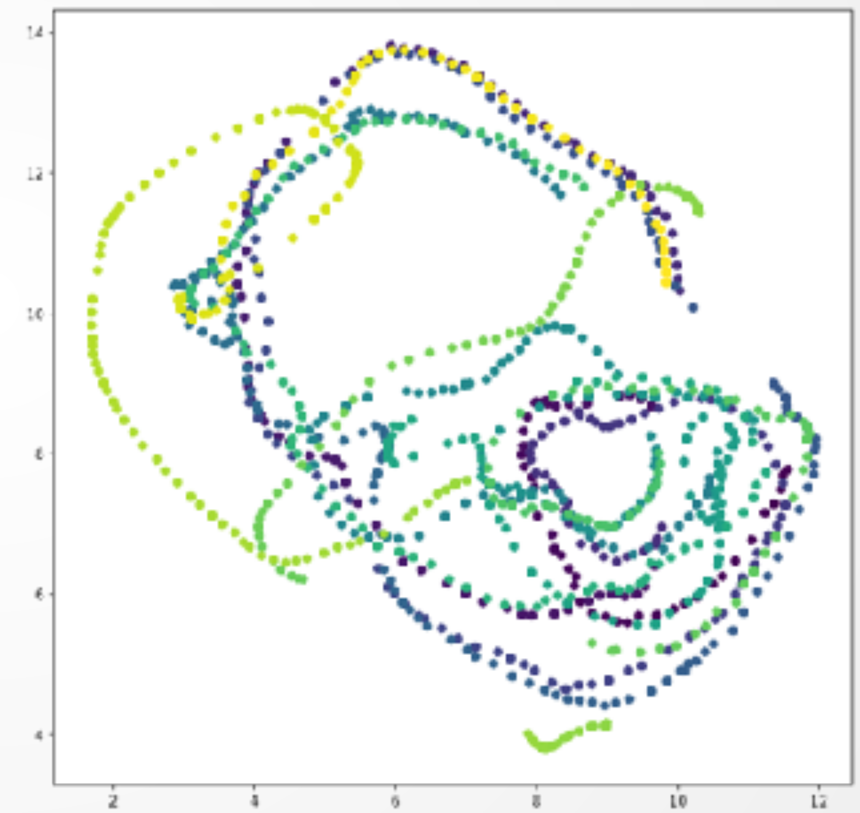


- UMAP recovers local structure for sufficient window size

Illustrative examples with simulated data

- Example 3: Only AR(1) process
- Autoregressive parameter: 0.9 Error sd: 0.1
- Window = lag size: *d=50*

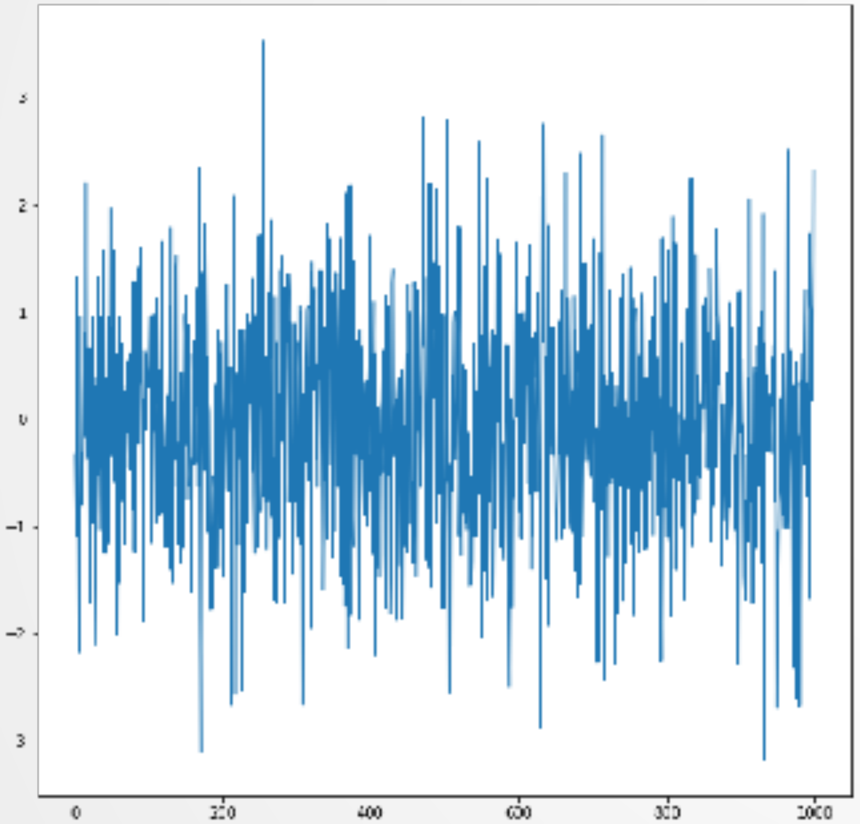


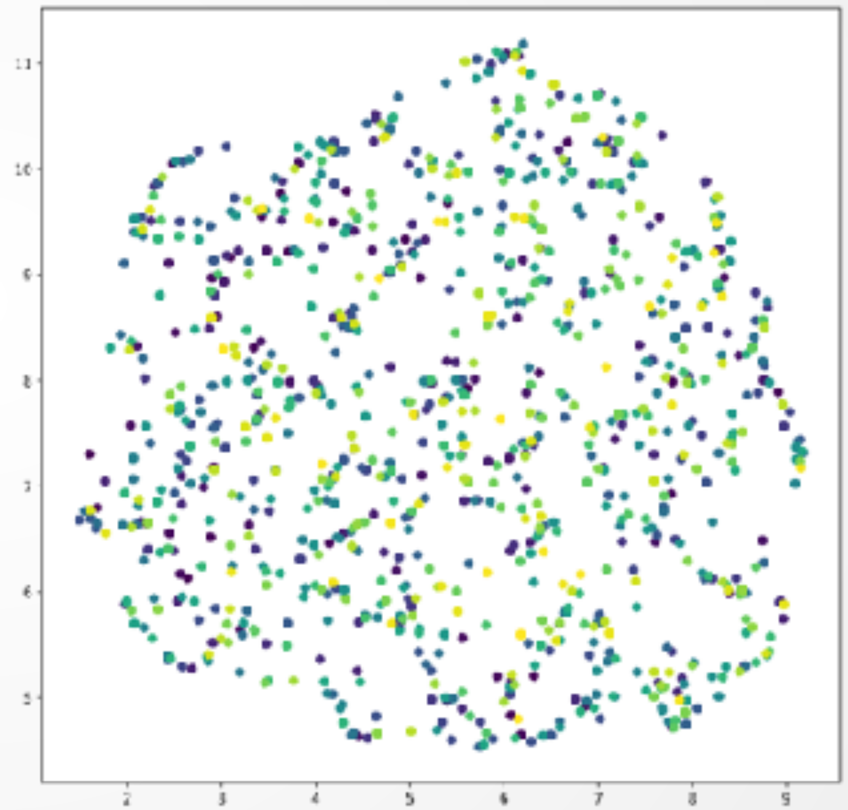


- UMAP recovers local structure for sufficient window size

Illustrative examples with simulated data

- Example 4: White Noise
- Gaussian White Noise: Sd = 1, Mean = 0
- Window = lag size: *d=5*

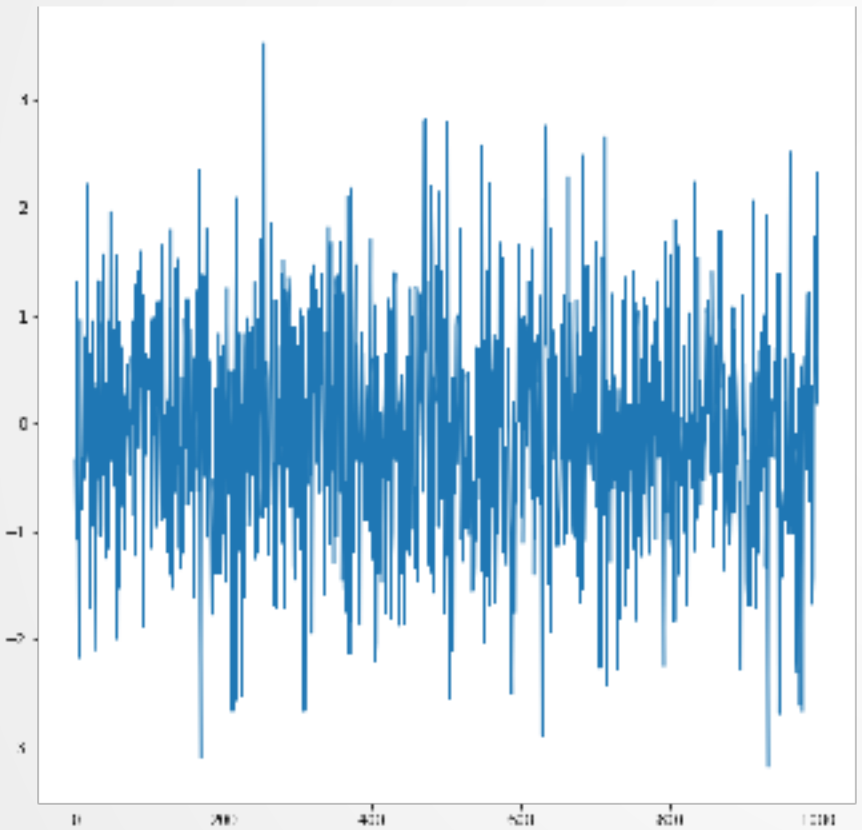


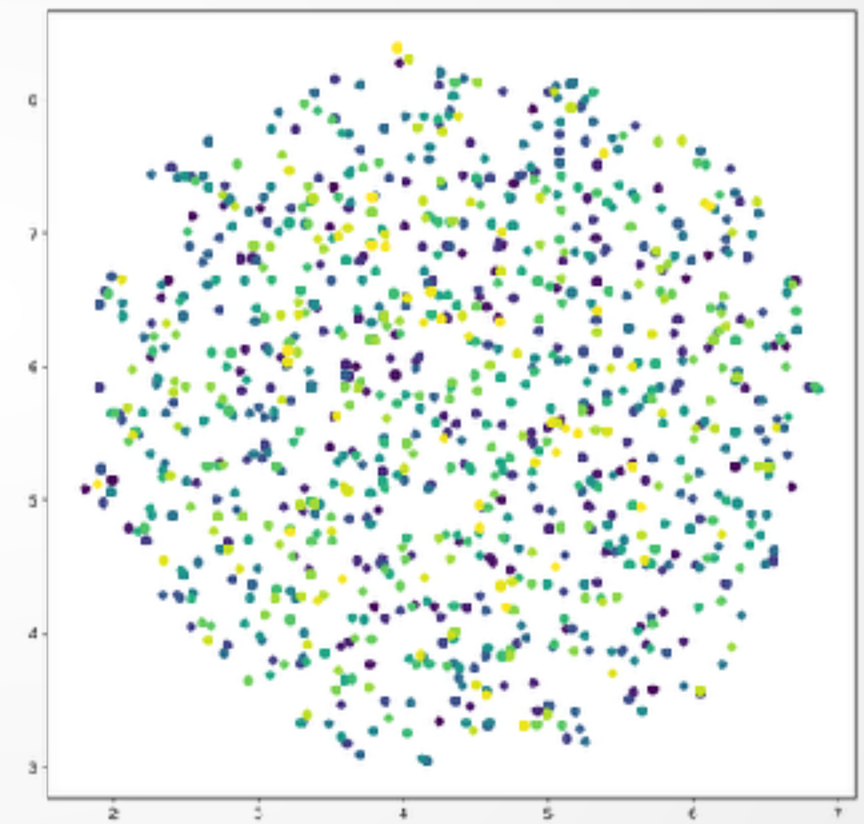


- UMAP recovers no structure

Illustrative examples with simulated data

- Example 4: White Noise
- Gaussian White Noise: Sd = 1, Mean = 0
- Window = lag size: *d=30*

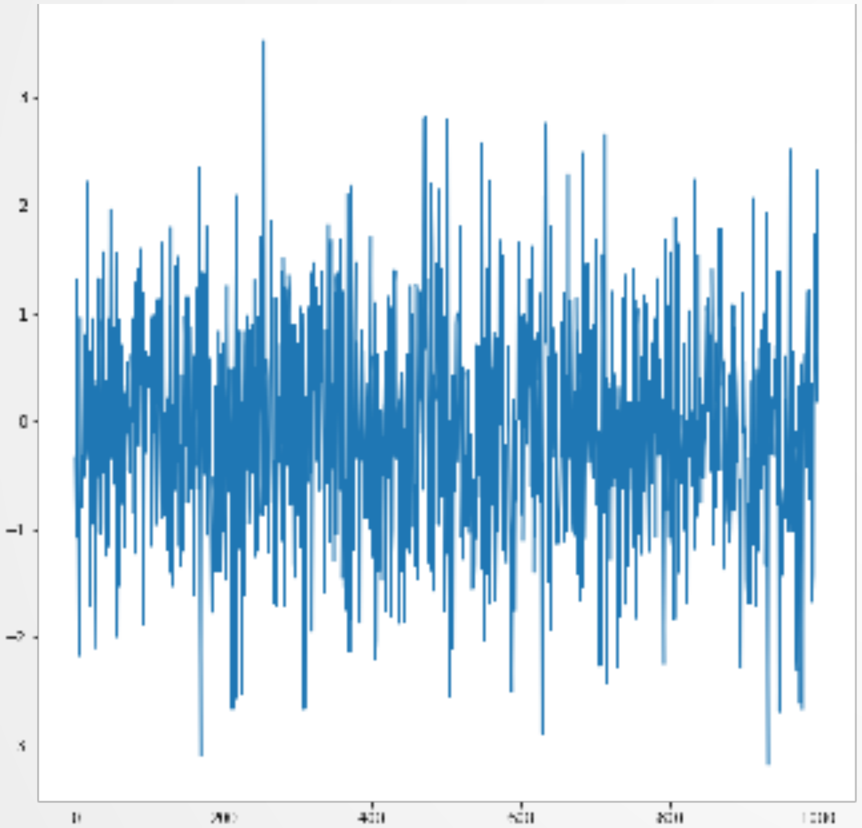


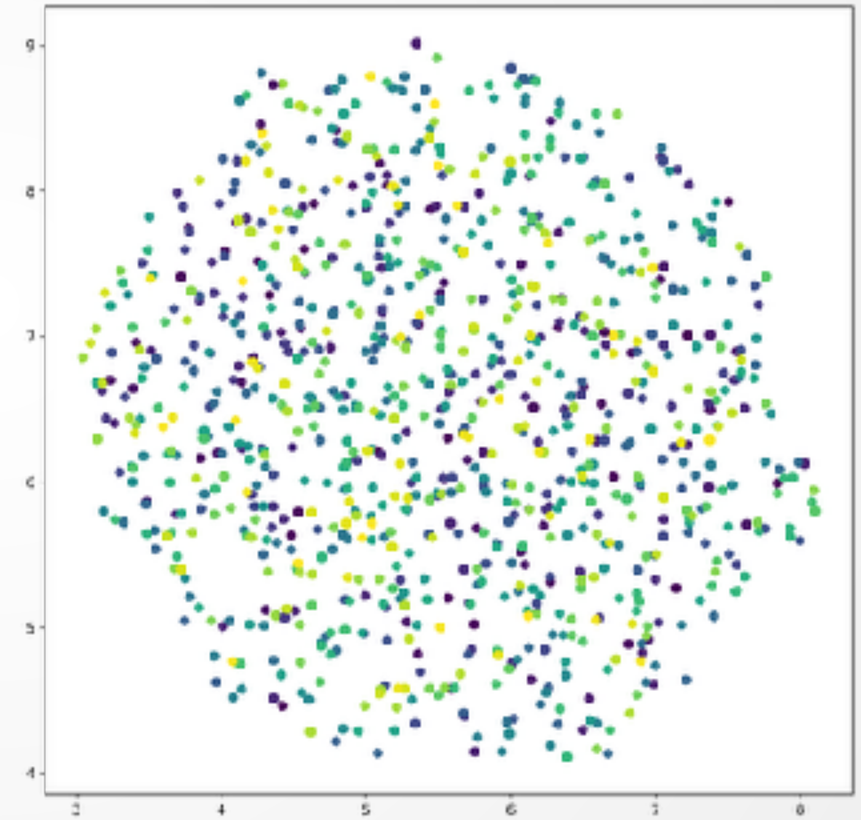


- UMAP recovers no structure

Illustrative examples with simulated data

- Example 4: White Noise
- Gaussian White Noise: Sd = 1, Mean = 0
- Window = lag size: *d=50*

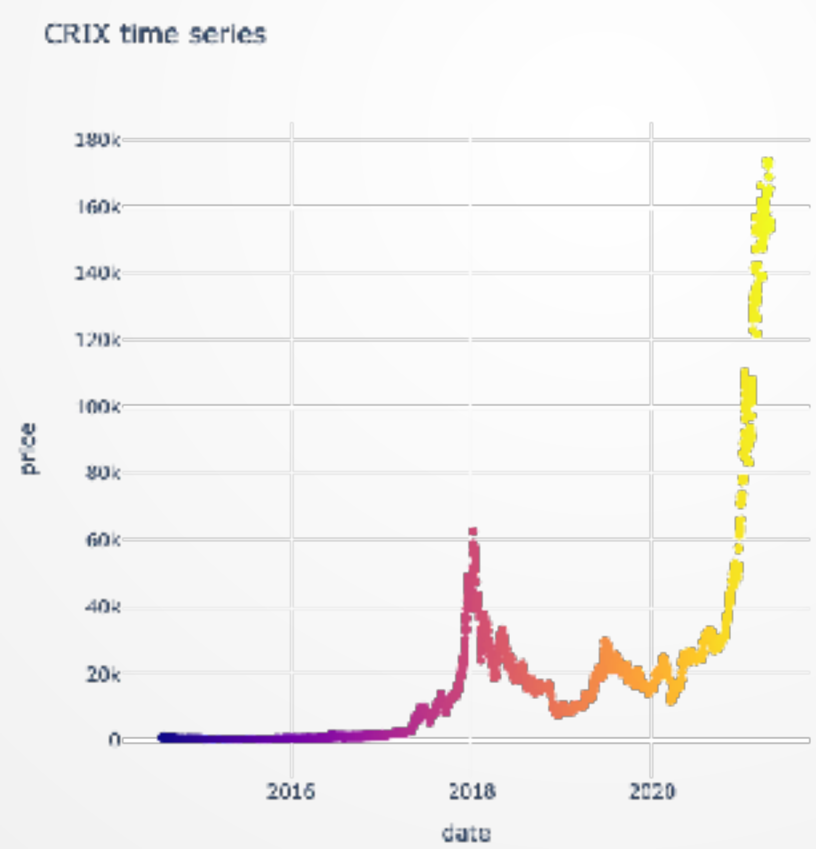




- UMAP recovers no structure

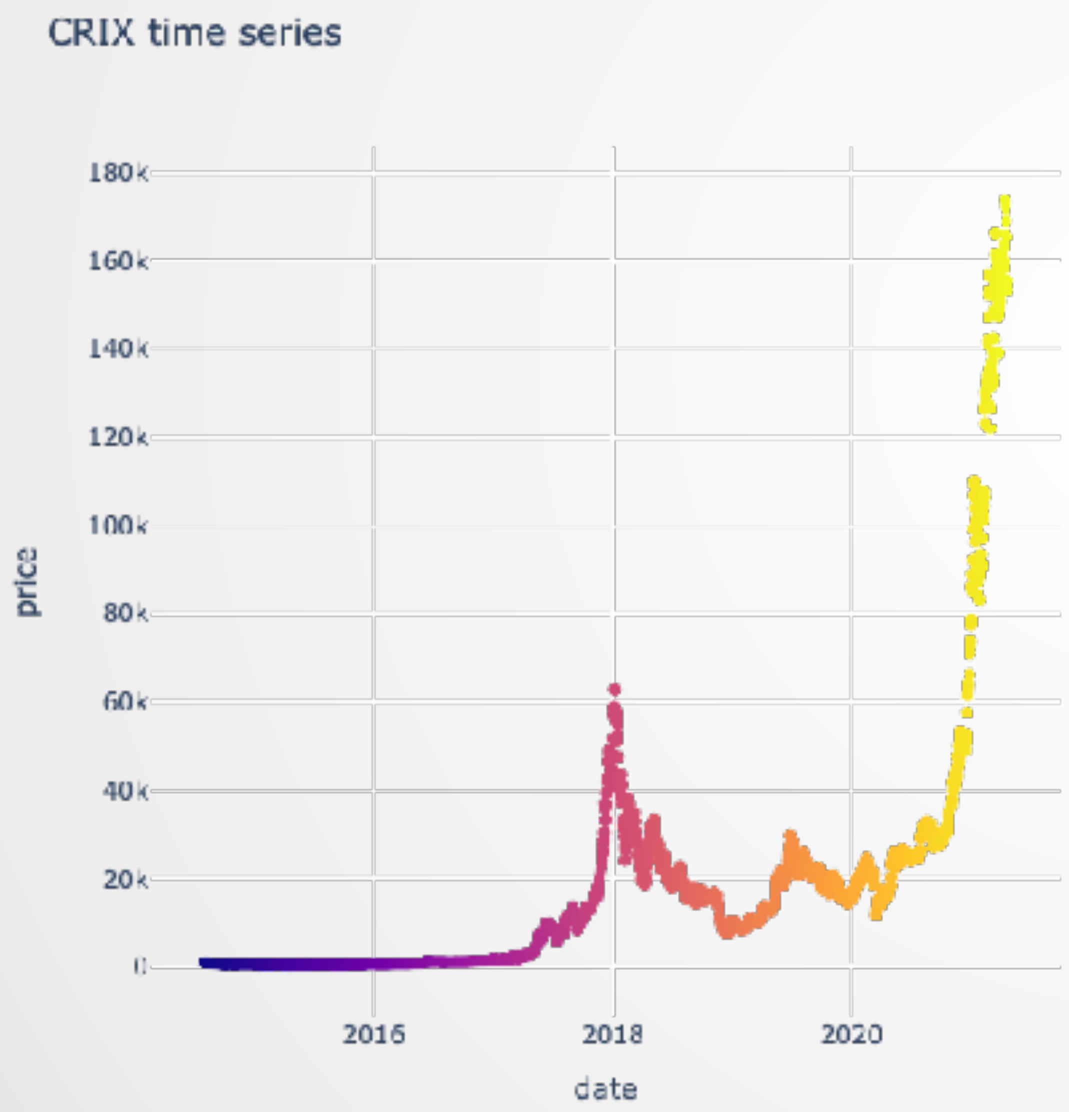
Time Series Example: CRIX

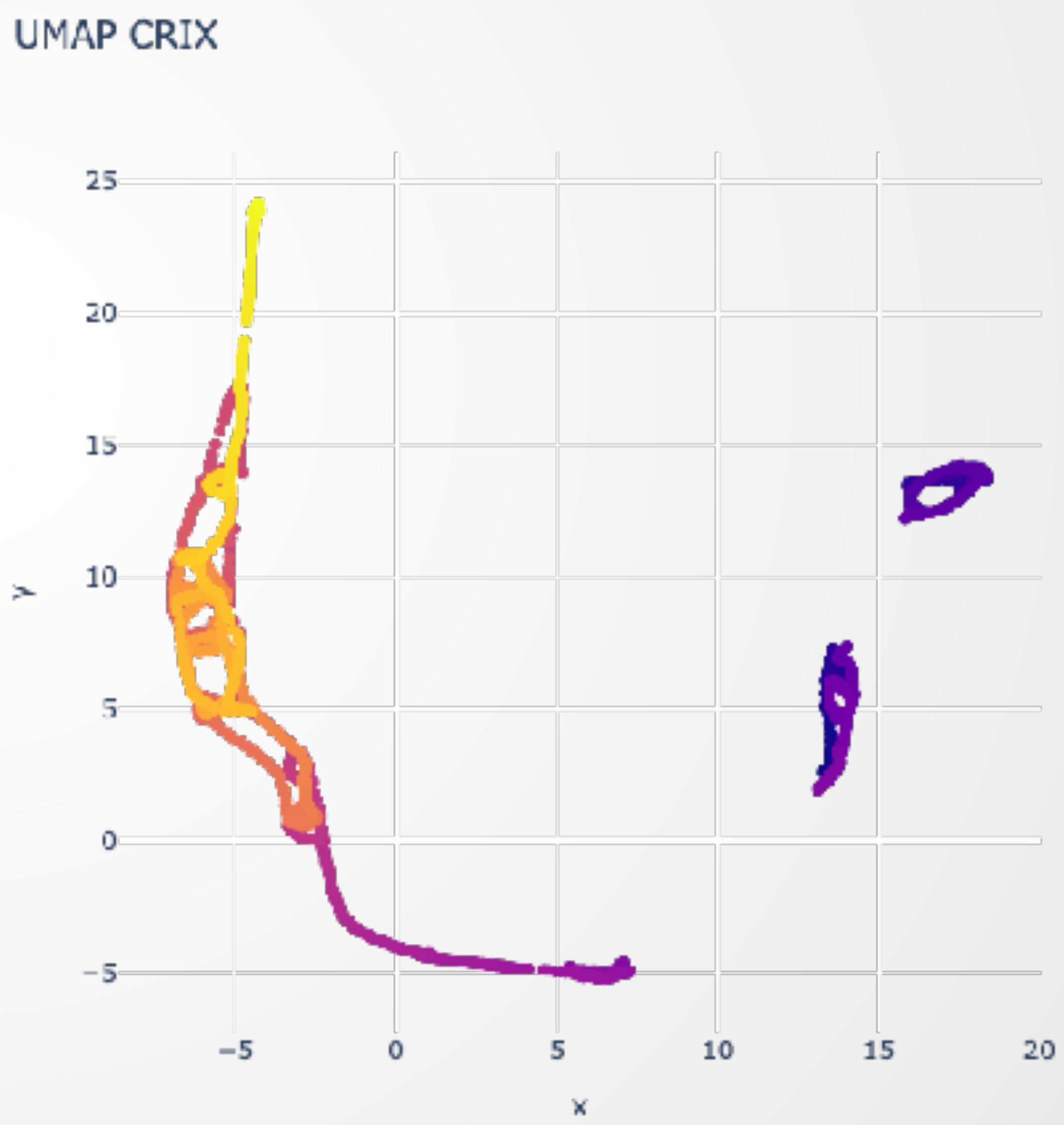
- CRIX is a market index for cryptocurrencies
- Constructed from cryptocurrencies with high market capitalization
- Captures dynamics of the cryptocurrencies market



Time Series Example: CRIX

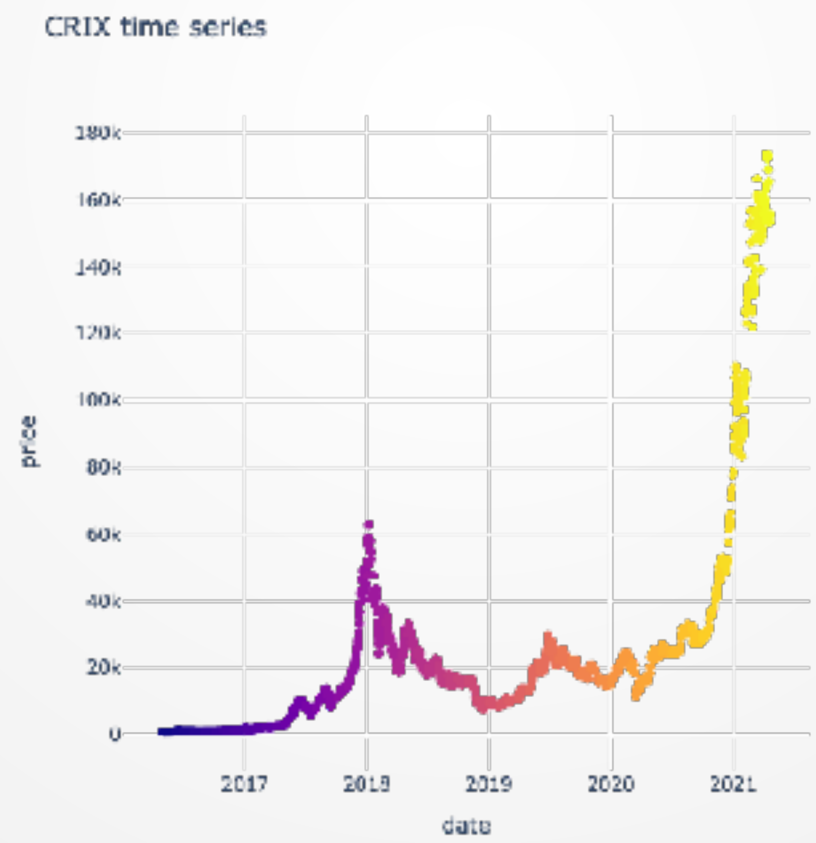
- Window size = 30, n_neighbors = 60



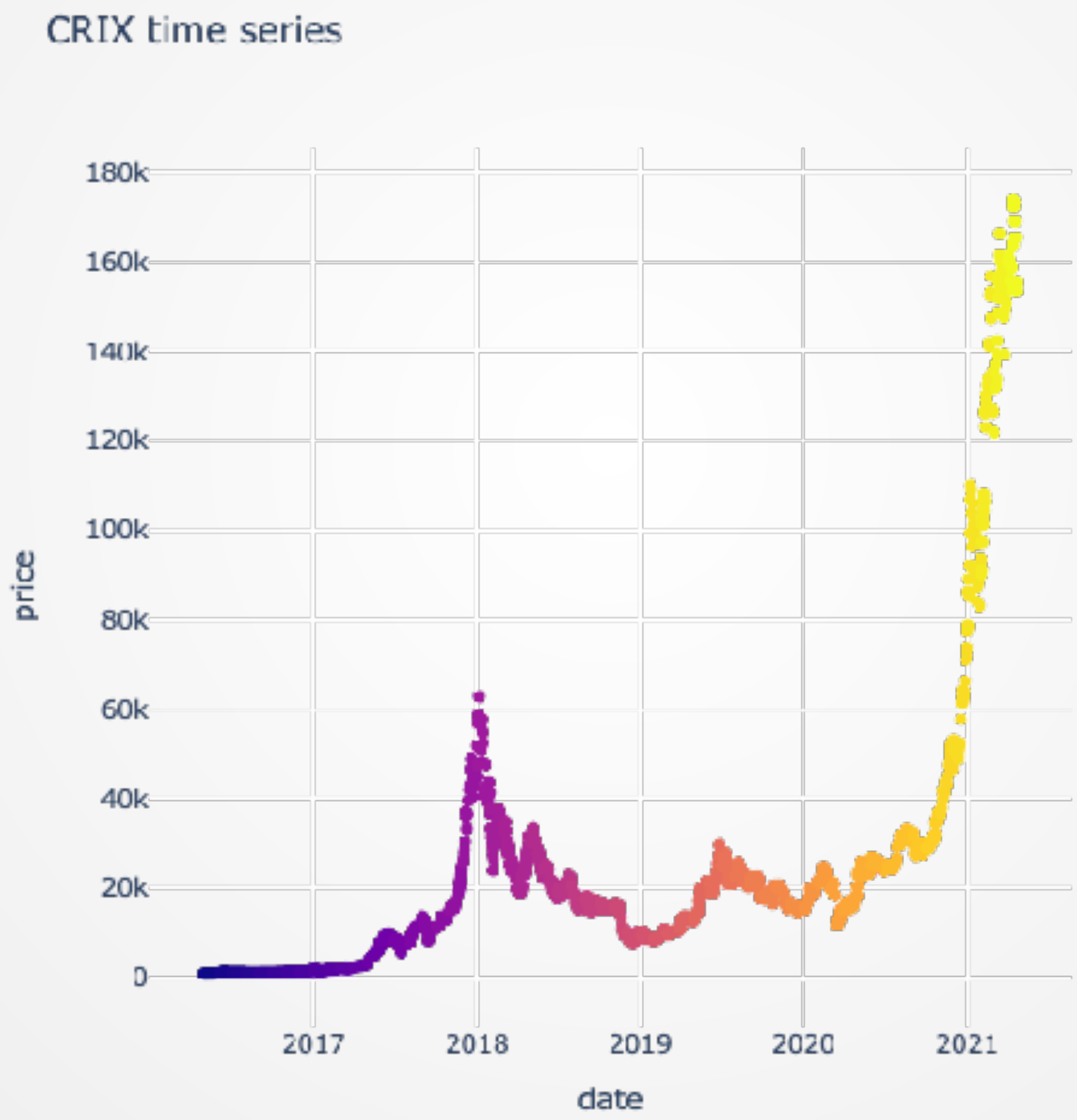


Time Series Example: CRIX

- Alternative method: look at constituents of CRIX instead of at CRIX time series
- Weights of CRIX constituents are a high-dimensional time series

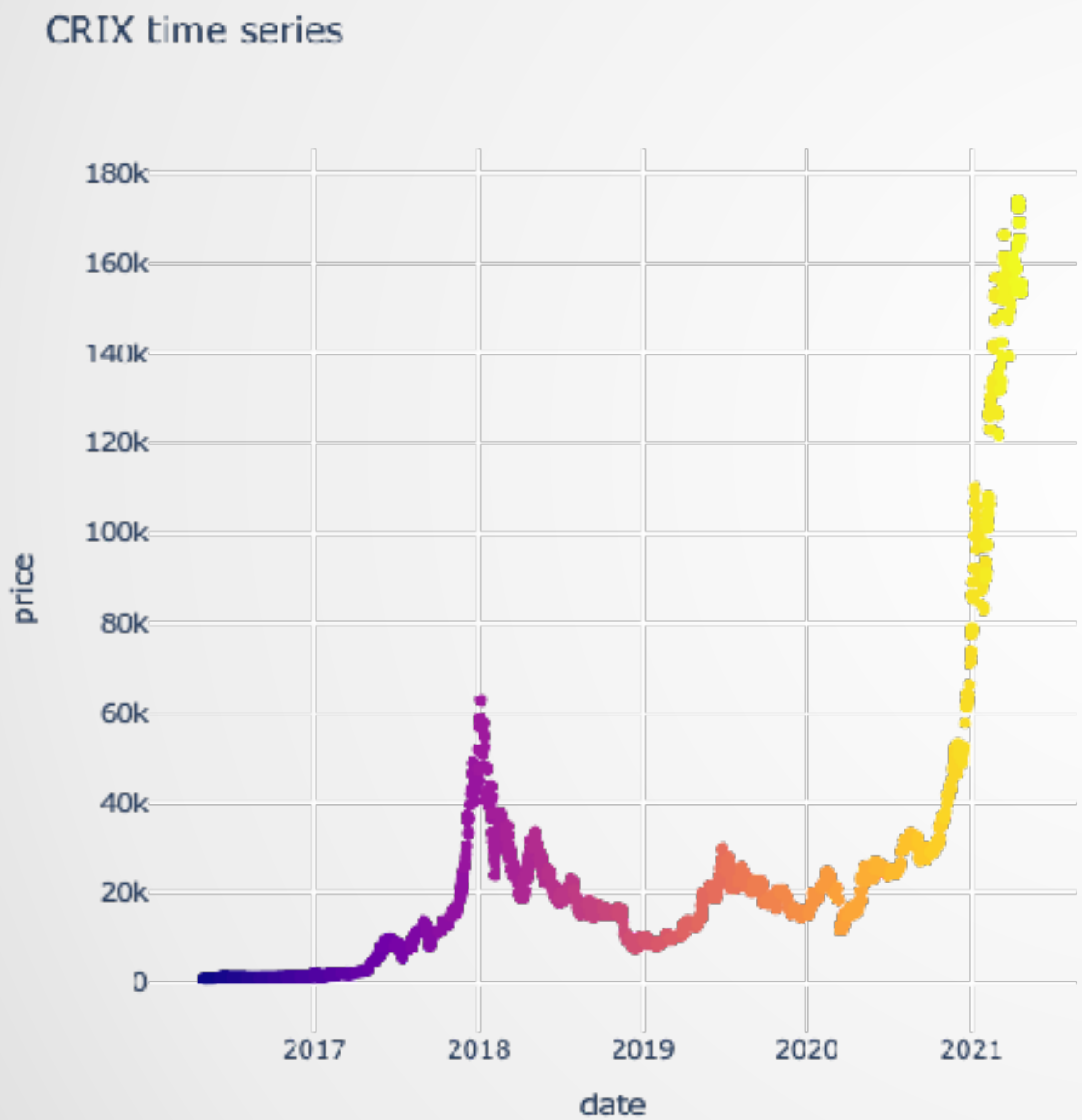


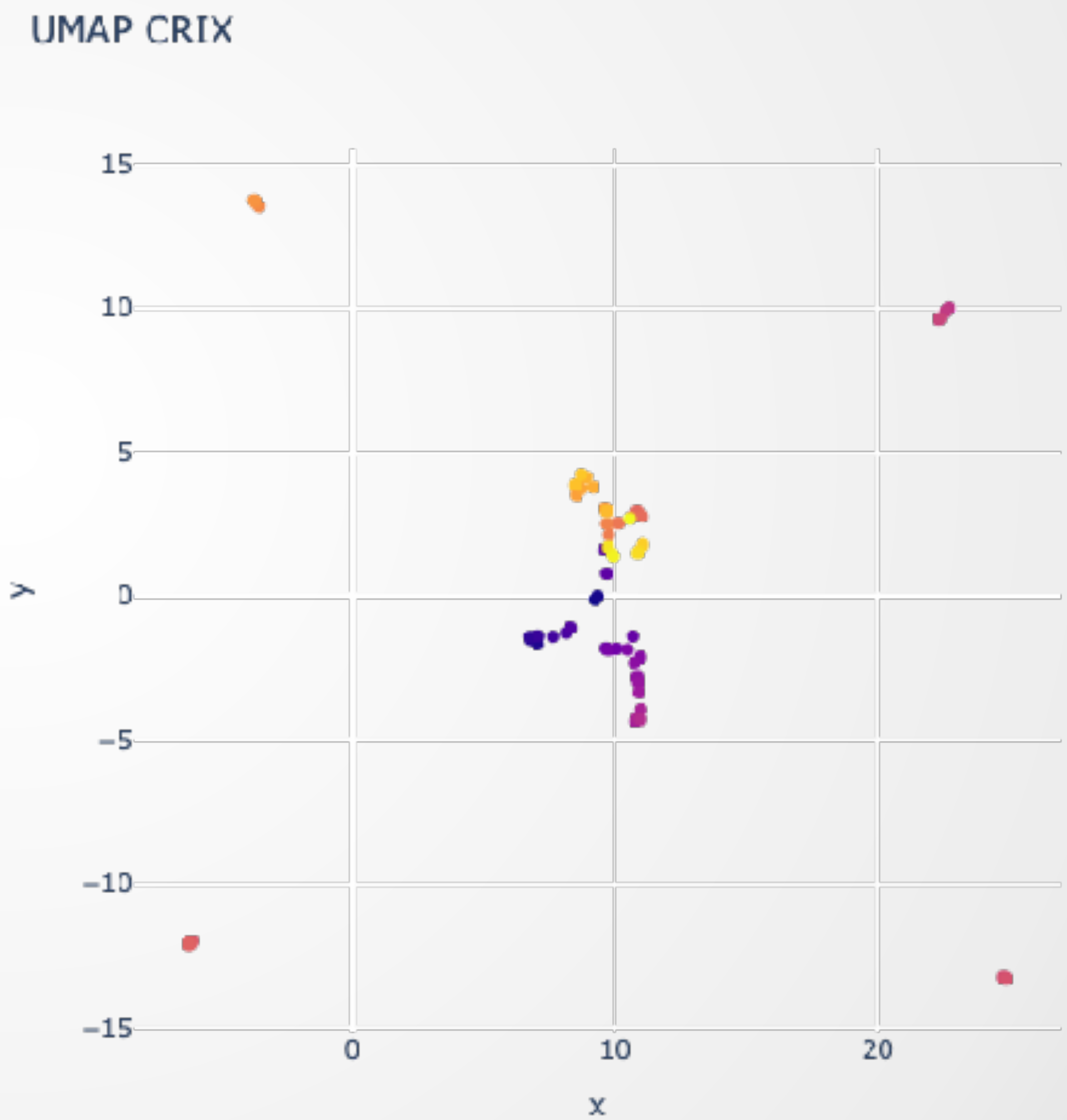
Time Series Example: CRIX



Time Series Example: CRIX

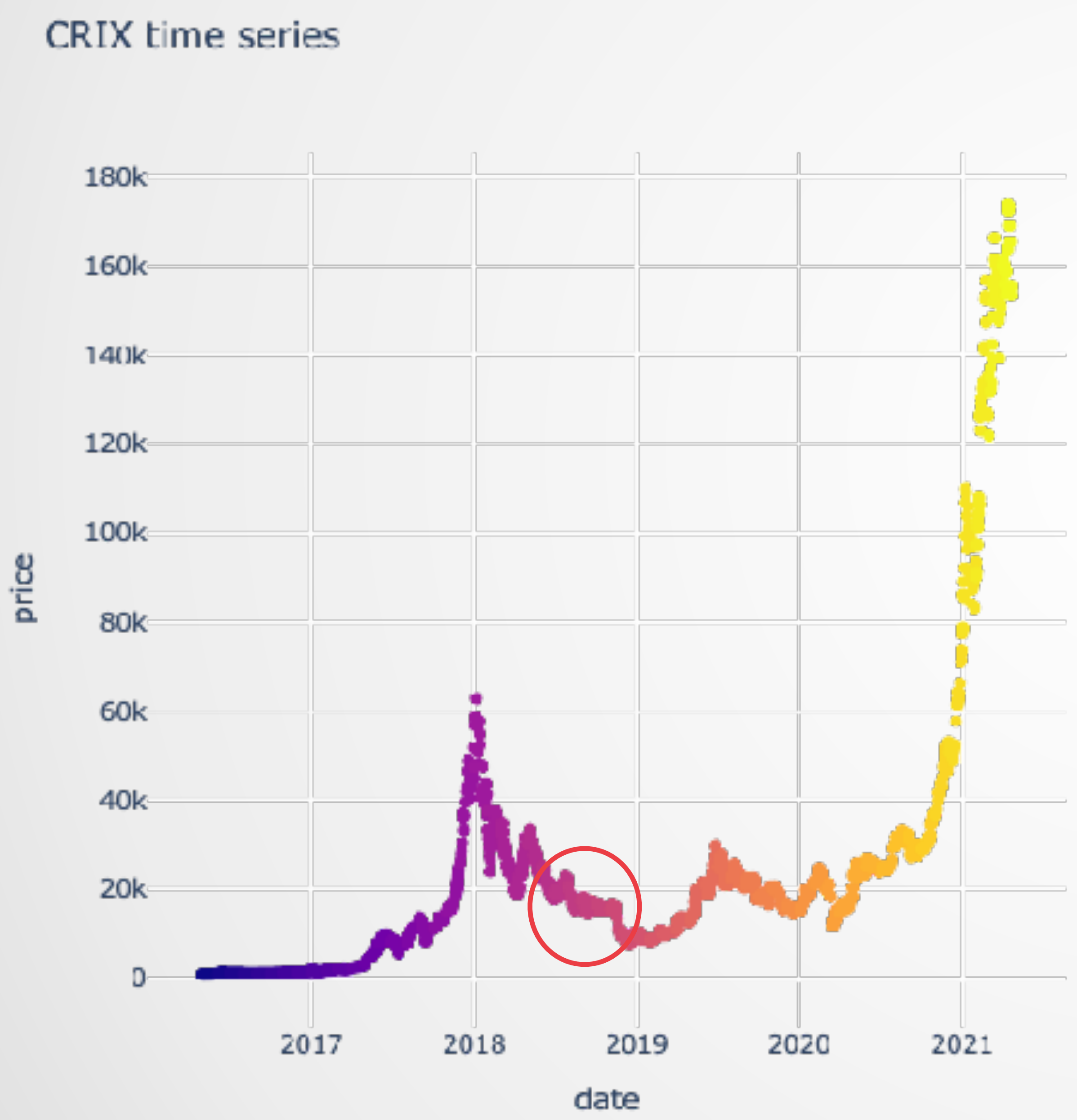
- n_neighbors = 3

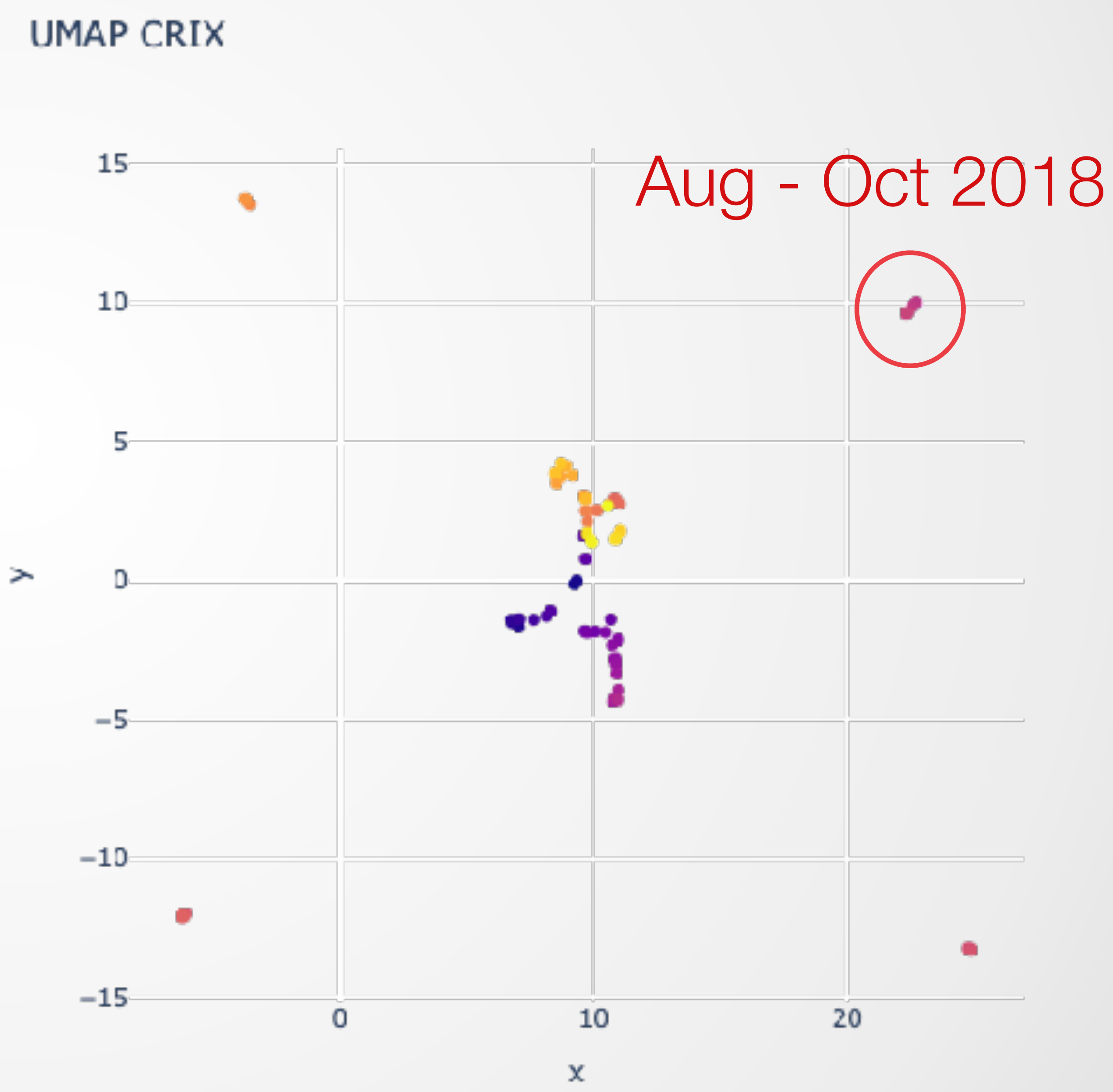




Time Series Example: CRIX

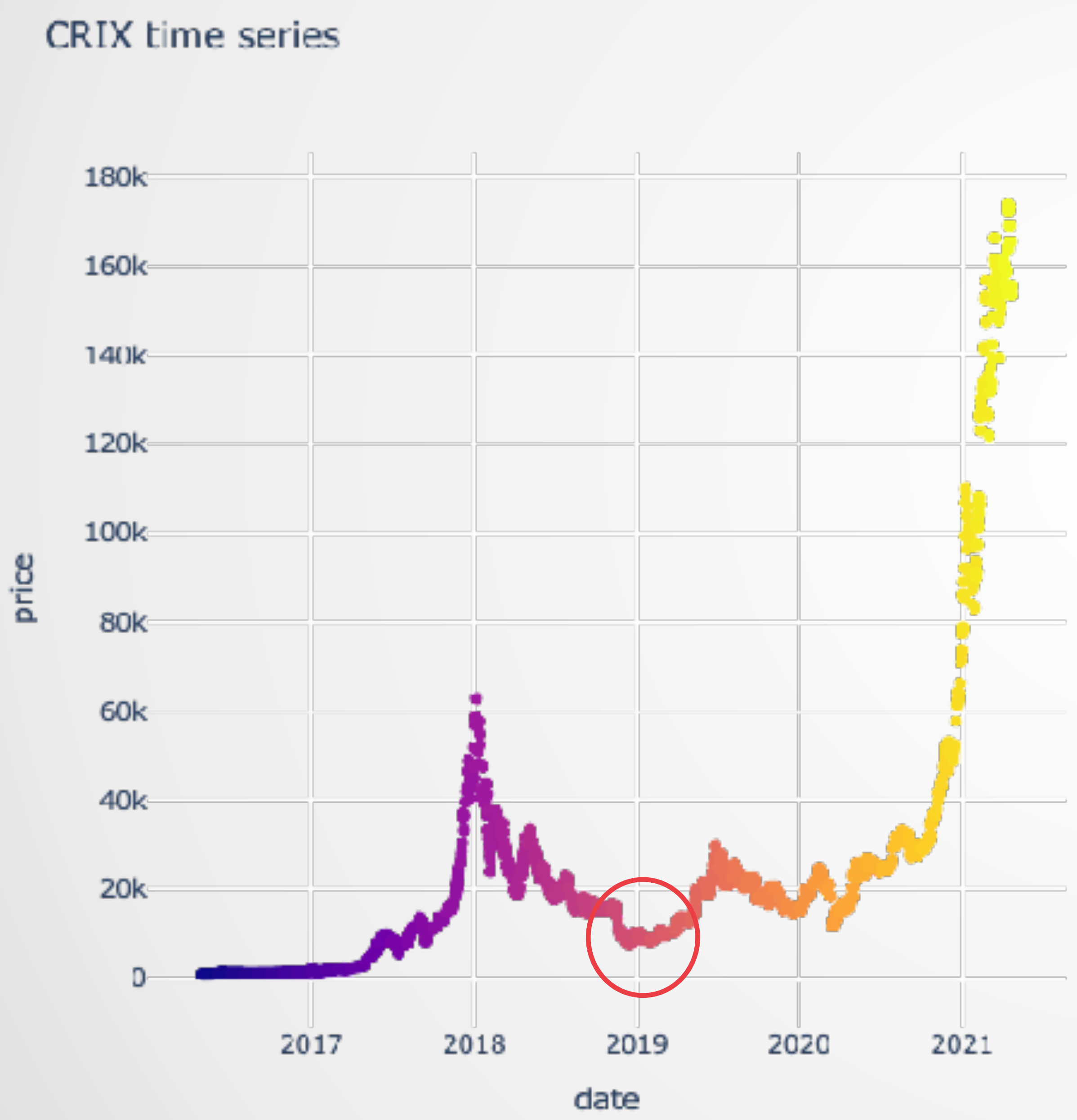
- n_neighbors = 3

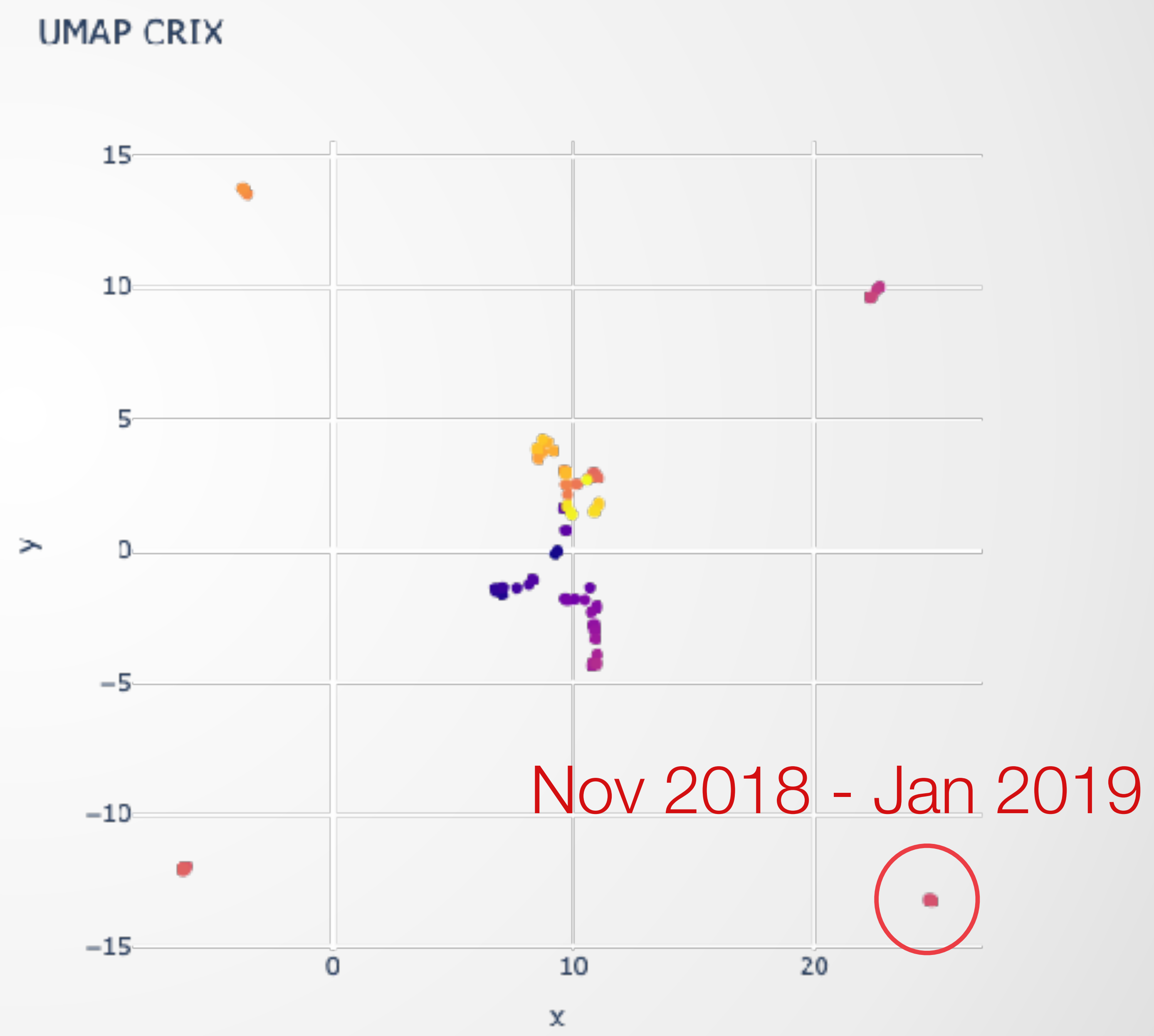




Time Series Example: CRIX

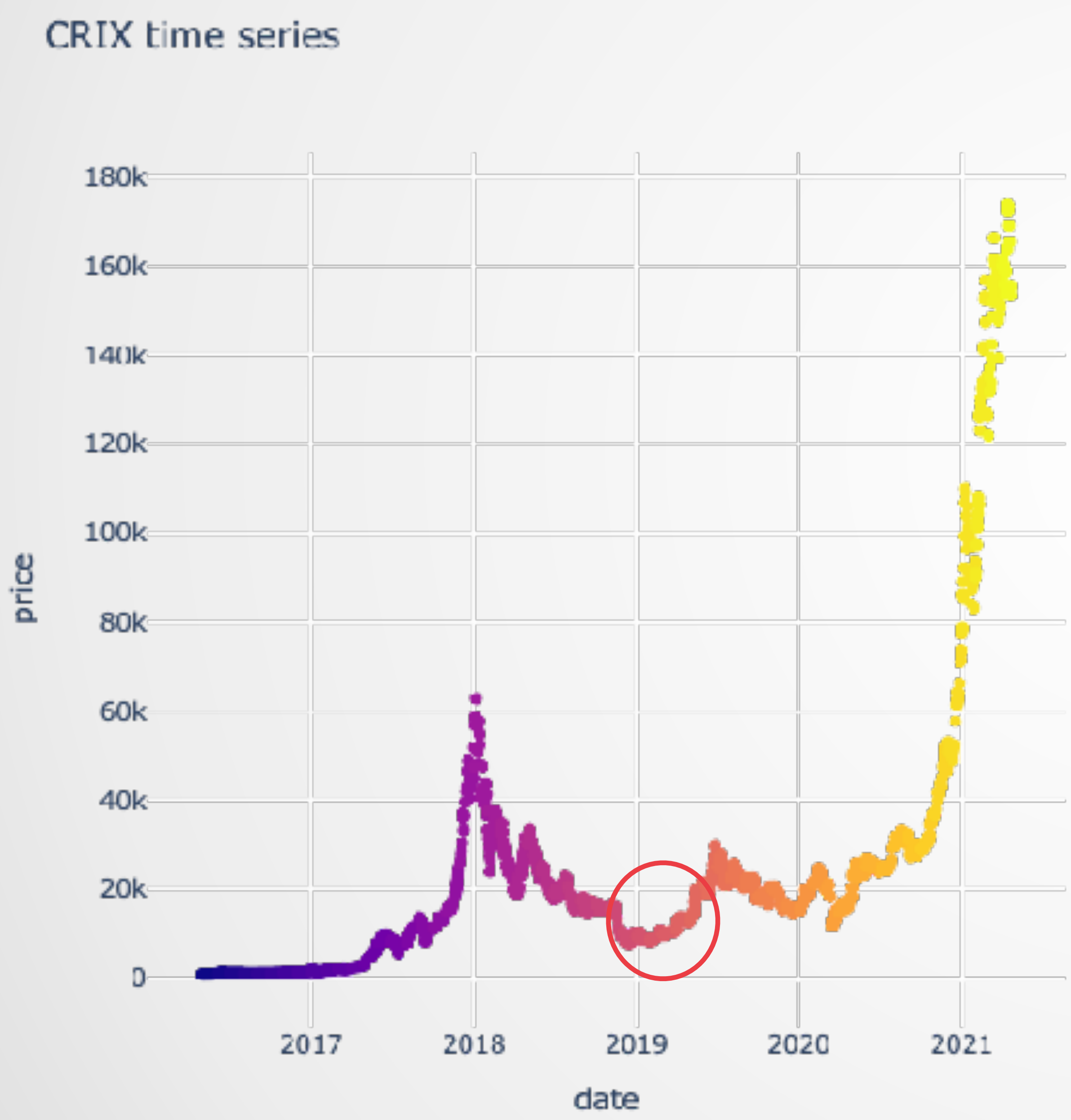
- n_neighbors = 3

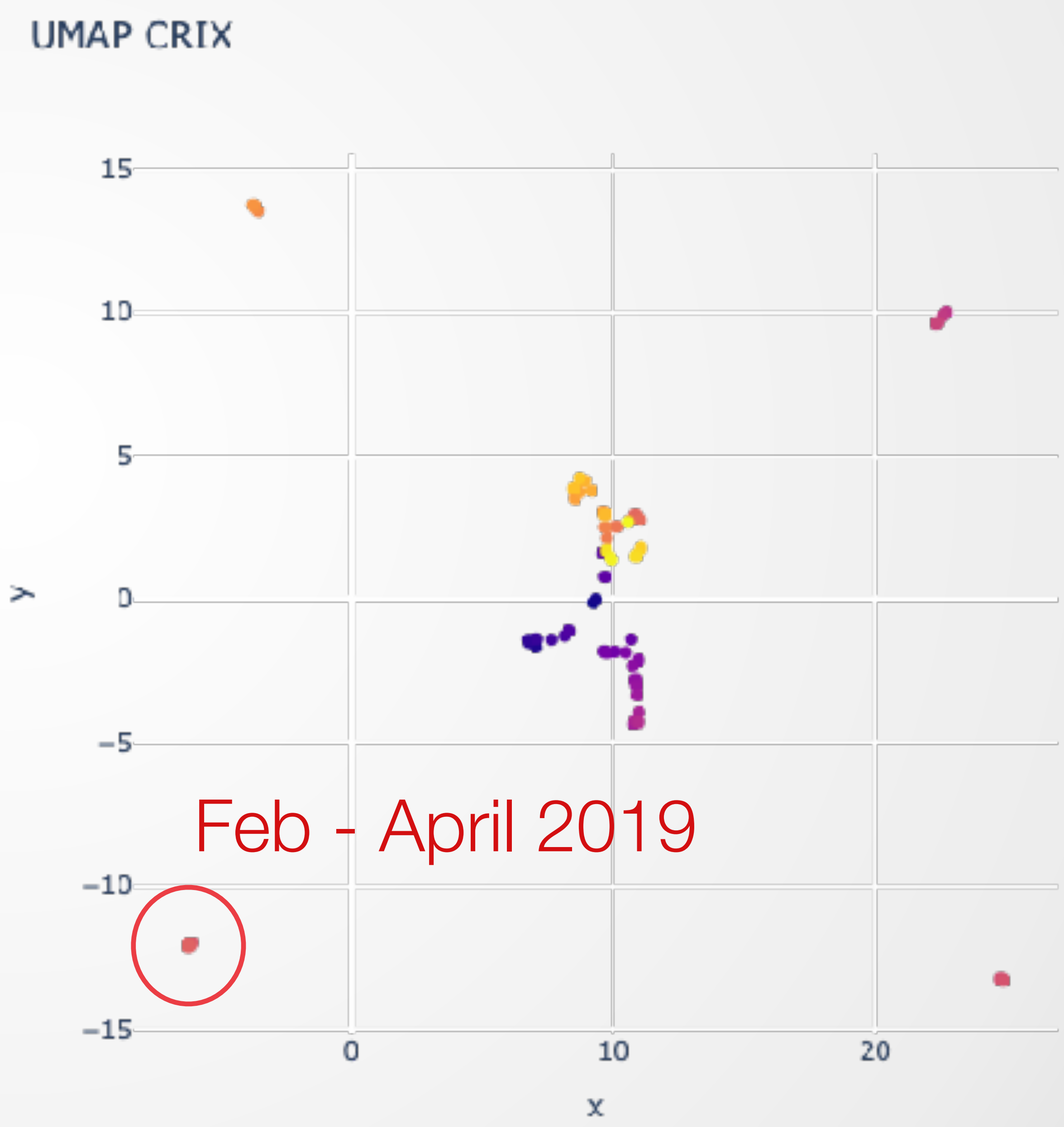




Time Series Example: CRIX

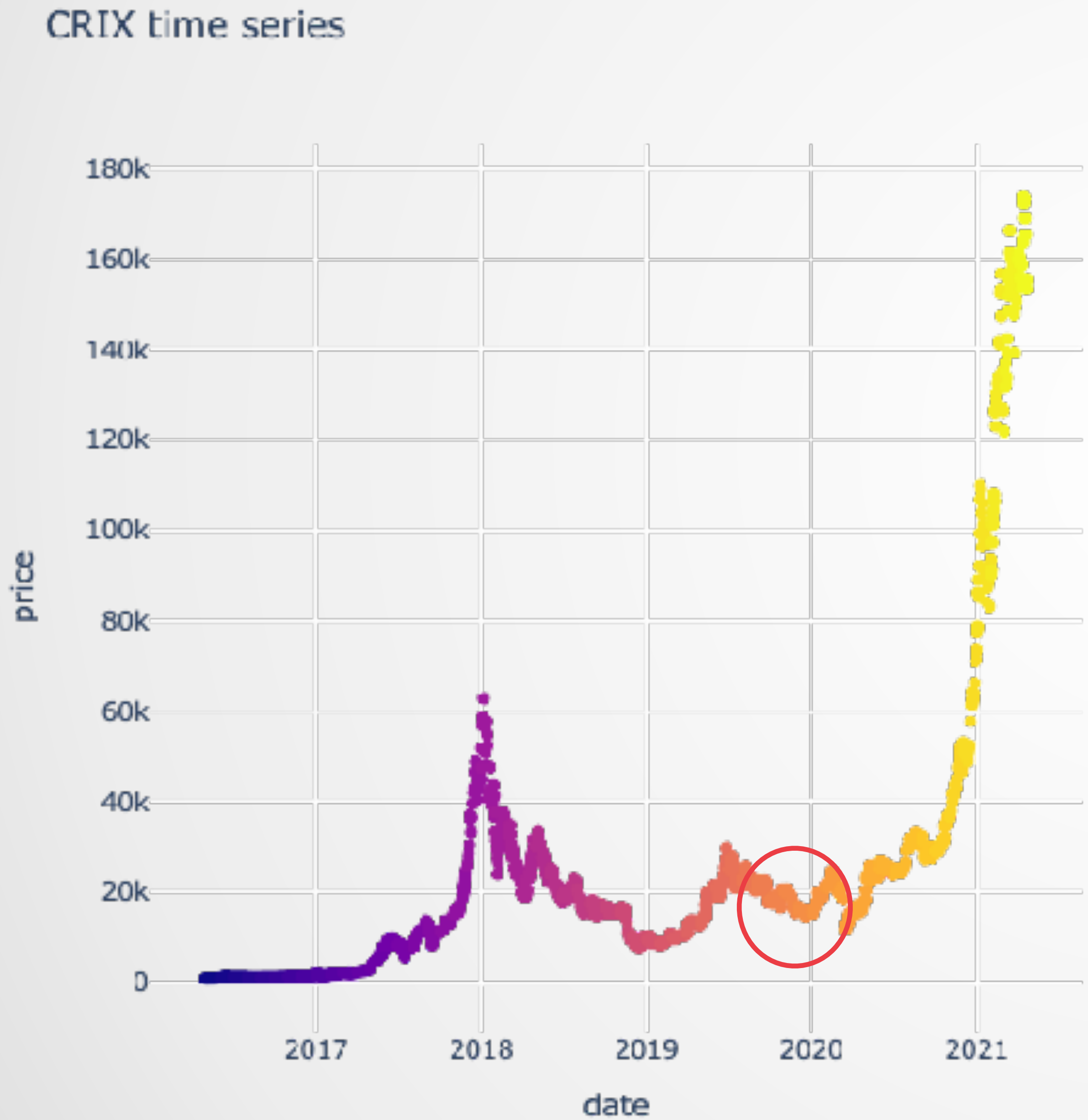
- n_neighbors = 3

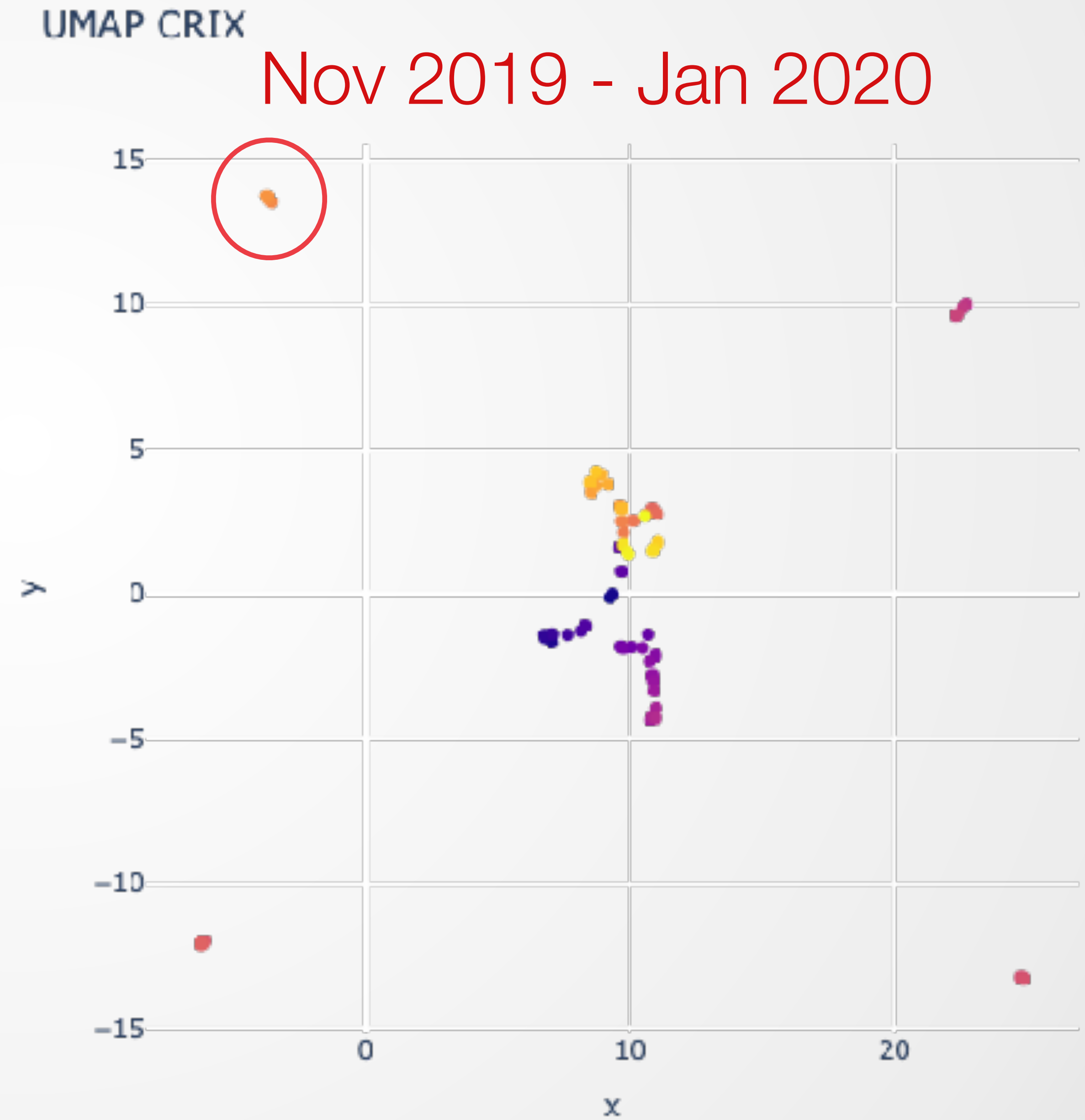




Time Series Example: CRIX

- n_neighbors = 3

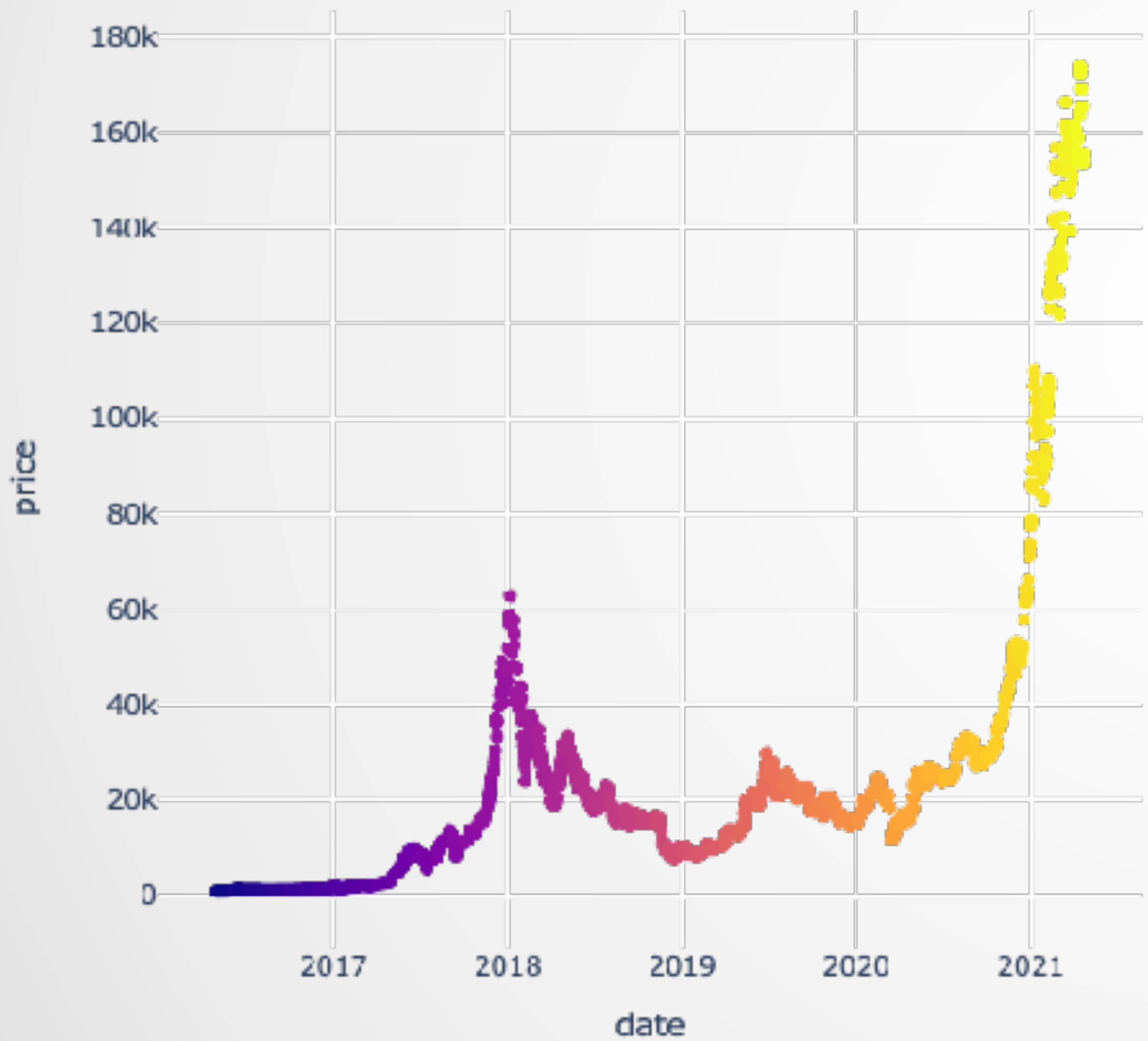




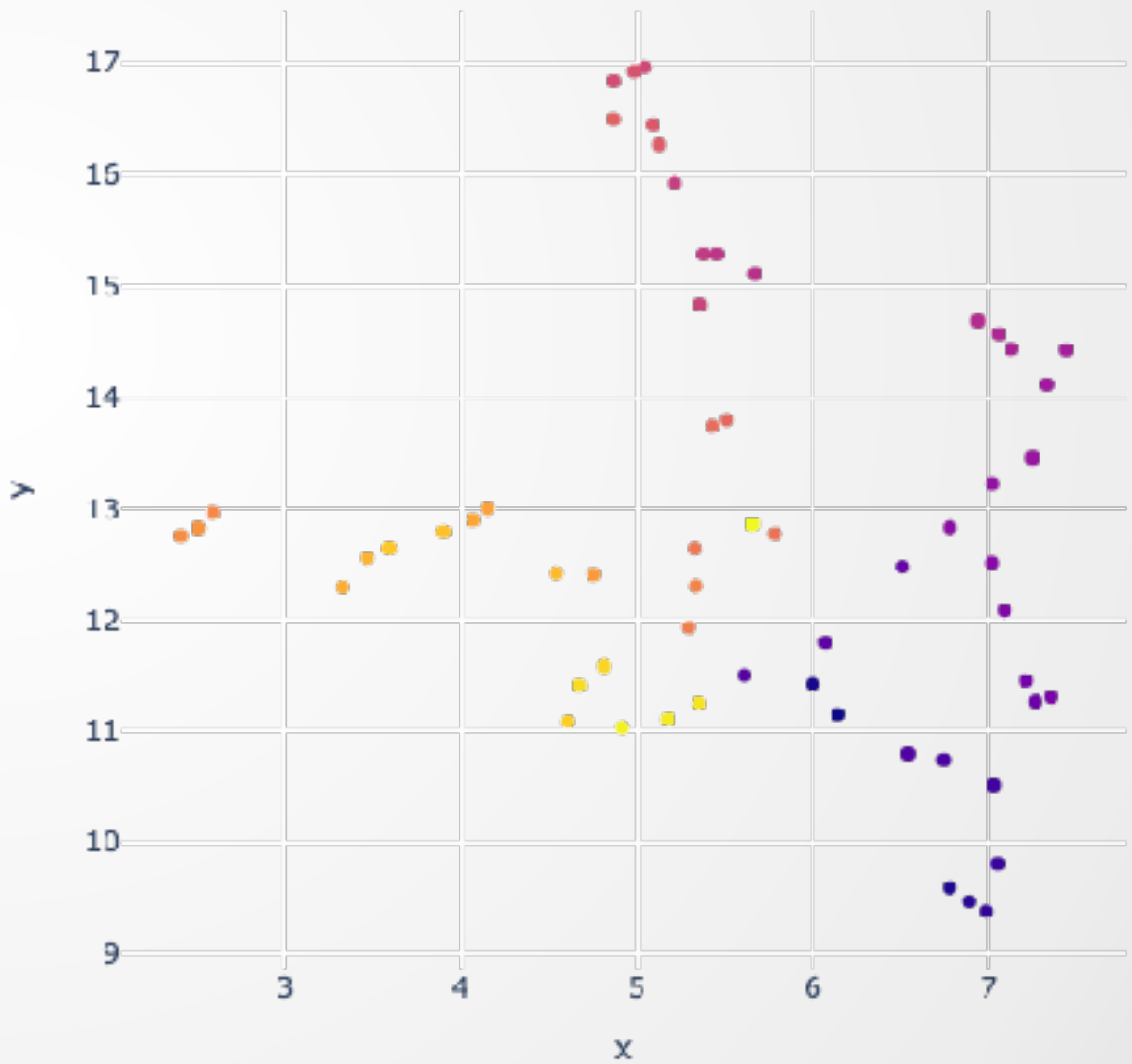
Time Series Example: CRIX

- n_neighbors = 5

CRIX time series

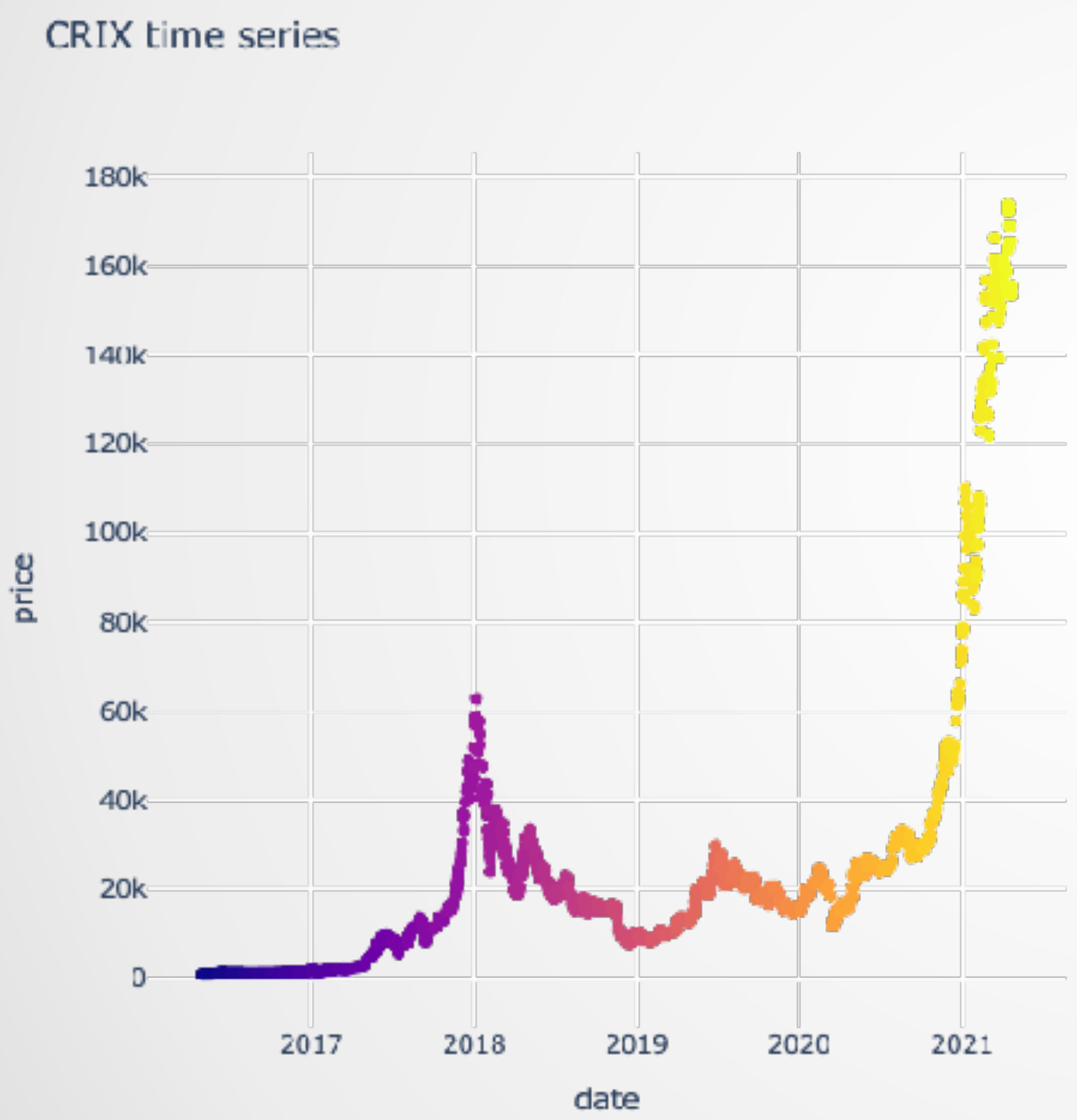


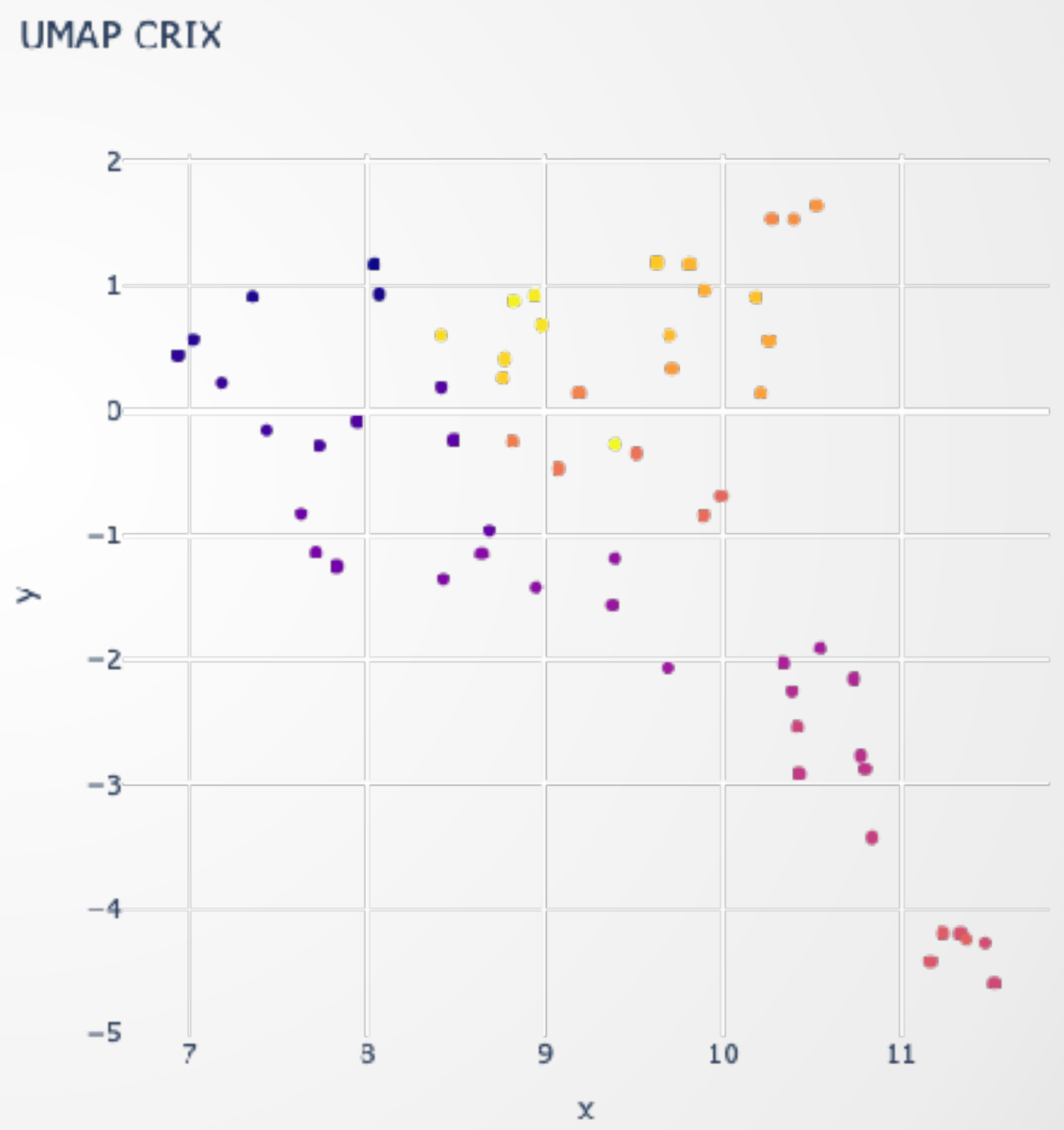
UMAP CRIX



Time Series Example: CRIX

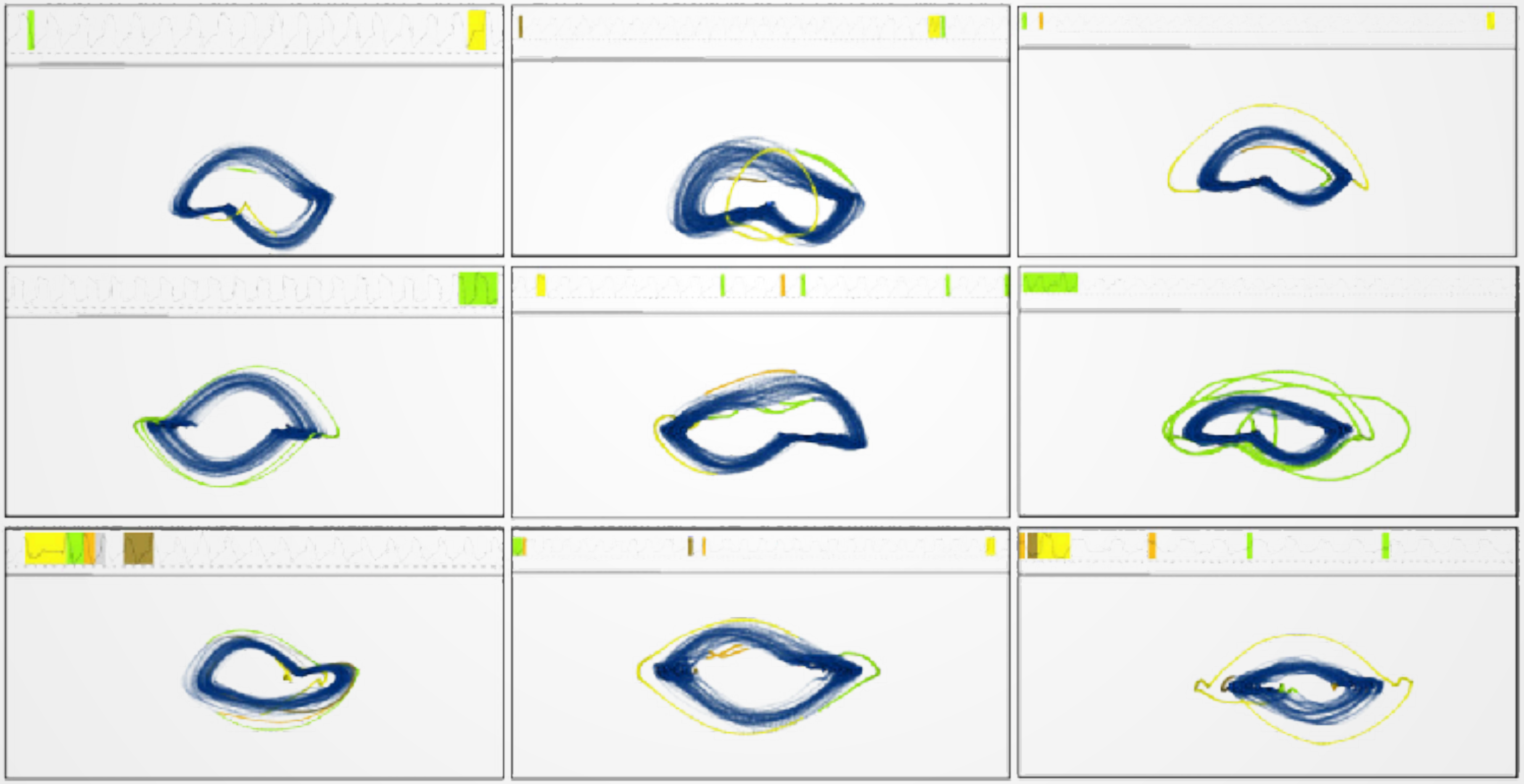
- n_neighbors = 12





Real data example: Ali, Jones, Xie et al. (2019)

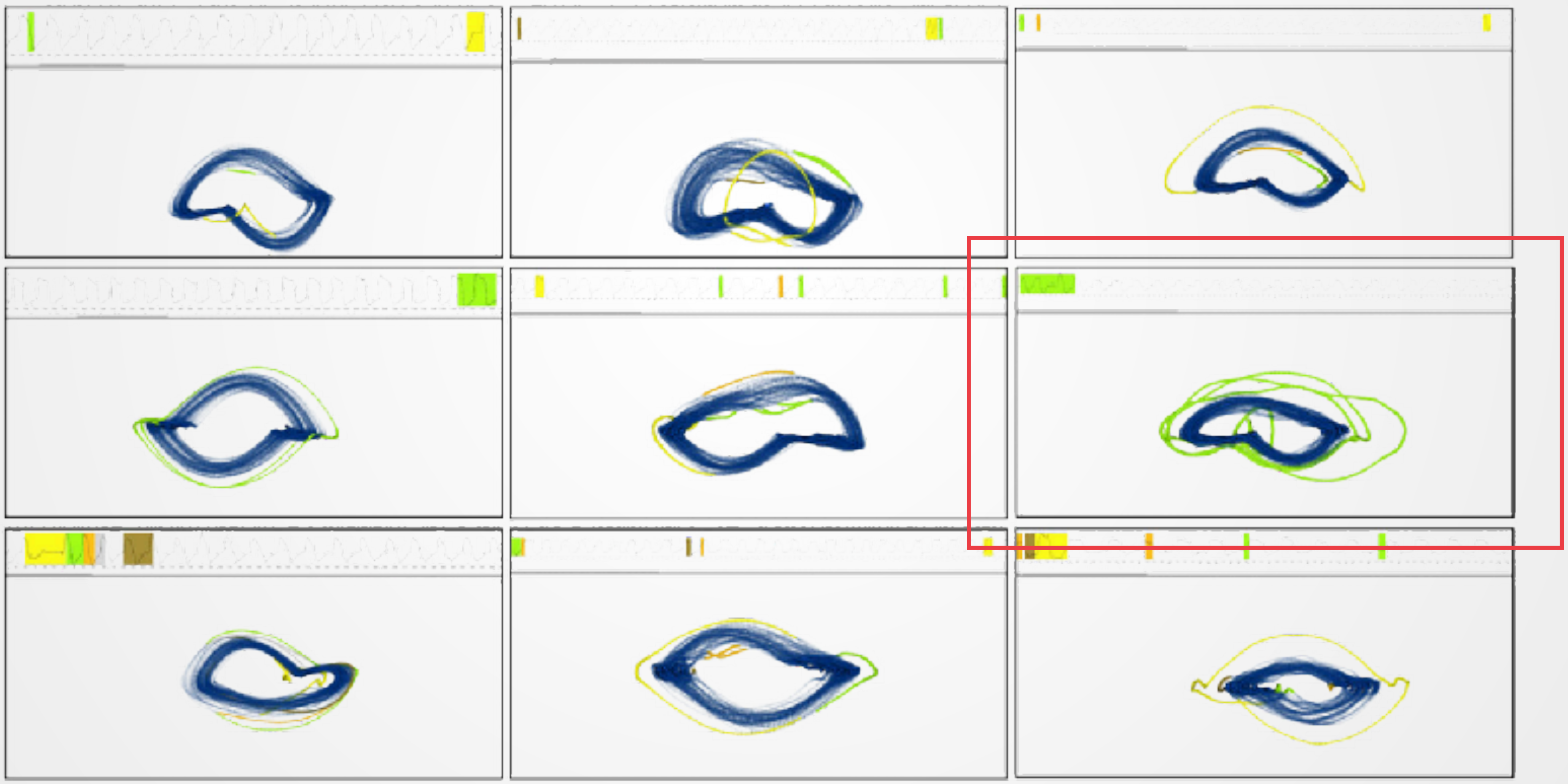
- UMAP of breathing patterns
- Diagnosis of problematic breathing patterns possible (Fig. 6)



Source: Ali, M., Jones, M.W., Xie, X. *et al.* TimeCluster: dimension reduction applied to temporal data for visual analytics. *Vis Comput* **35,** 1013–1026 (2019). https://doi.org/10.1007/s00371-019-01673-y

Real data example: Ali, Jones, Xie et al. (2019)

- UMAP of breathing patterns
- Diagnosis of problematic breathing patterns possible (Fig. 6)



Source: Ali, M., Jones, M.W., Xie, X. *et al.* TimeCluster: dimension reduction applied to temporal data for visual analytics. *Vis Comput* **35,** 1013–1026 (2019). https://doi.org/10.1007/s00371-019-01673-y

UMAP bottomline

- Useful dimension reduction technique
- Can serve as intermediate step to increase classification accuracy of high dimensional data
- Computationally fast
- Recovers local and global structure

Cost function

_t_-SNE: KL-divergence

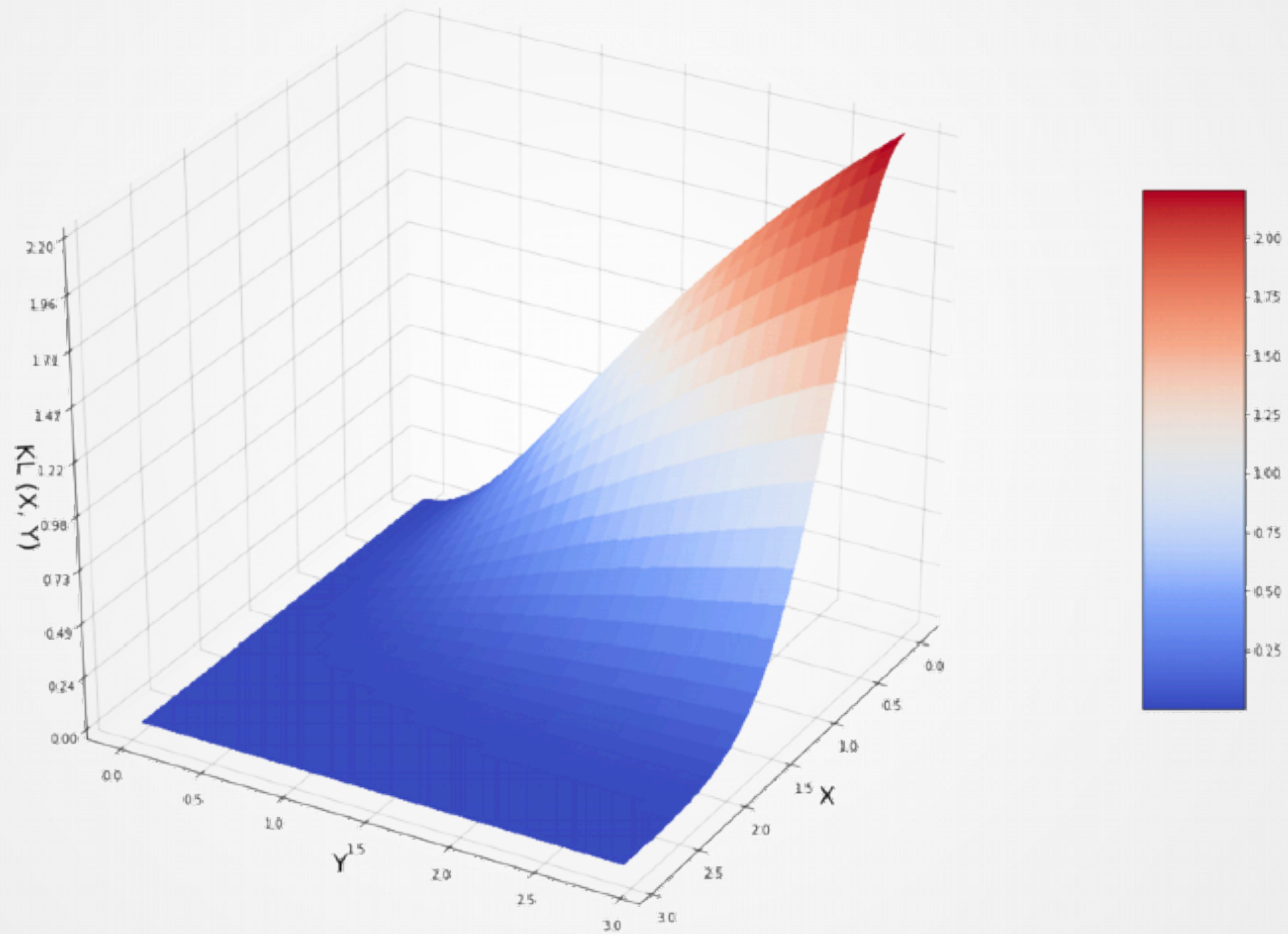
Impossible to preserve global distances

UMAP: Cross Entropy (CE)

Smarter cost function to preserve global distances

KL-divergence

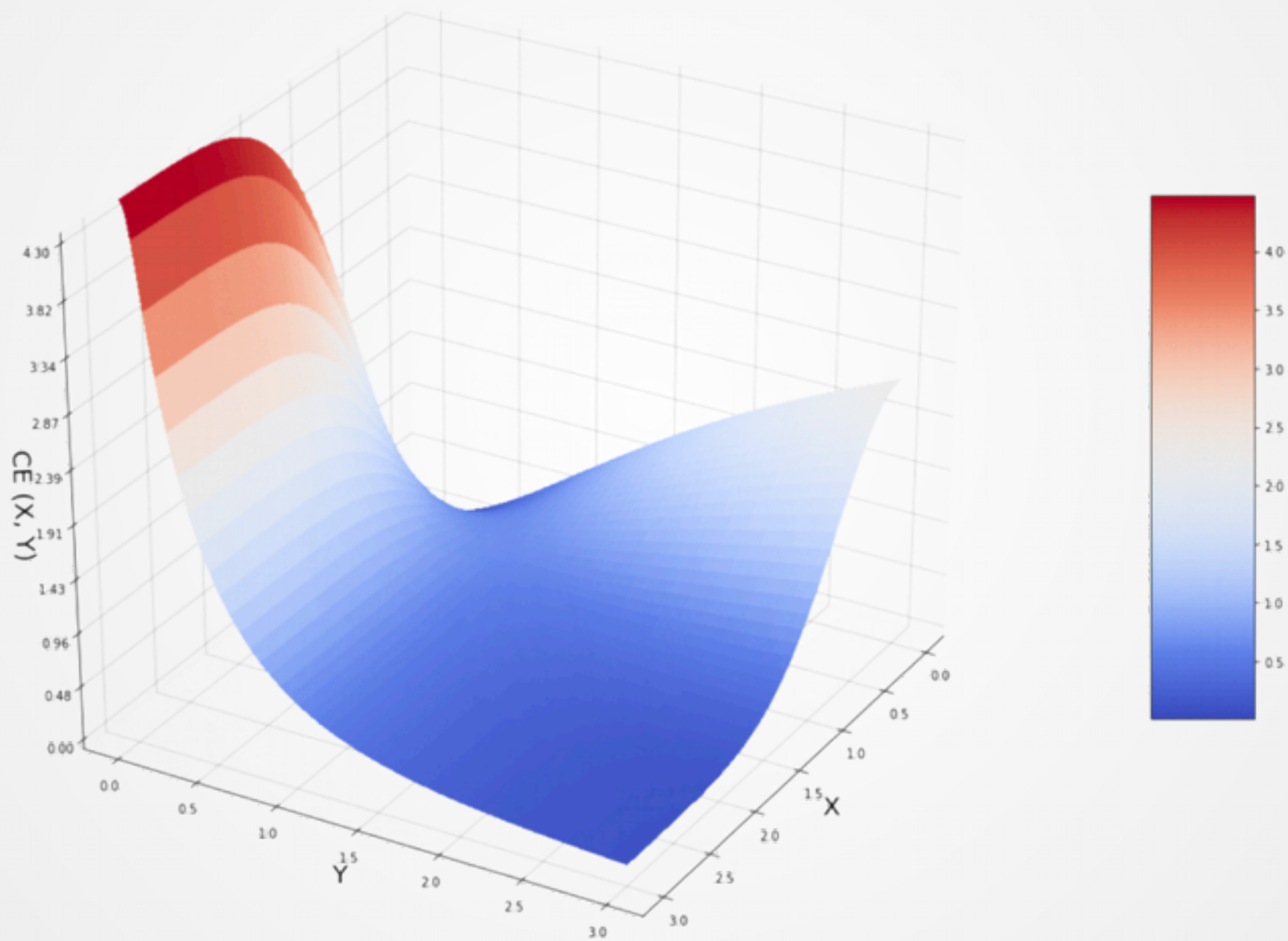
- Large X does not guarantee a large Y



X: distance between points in high dimensions   Y: distance between points in low dimensions

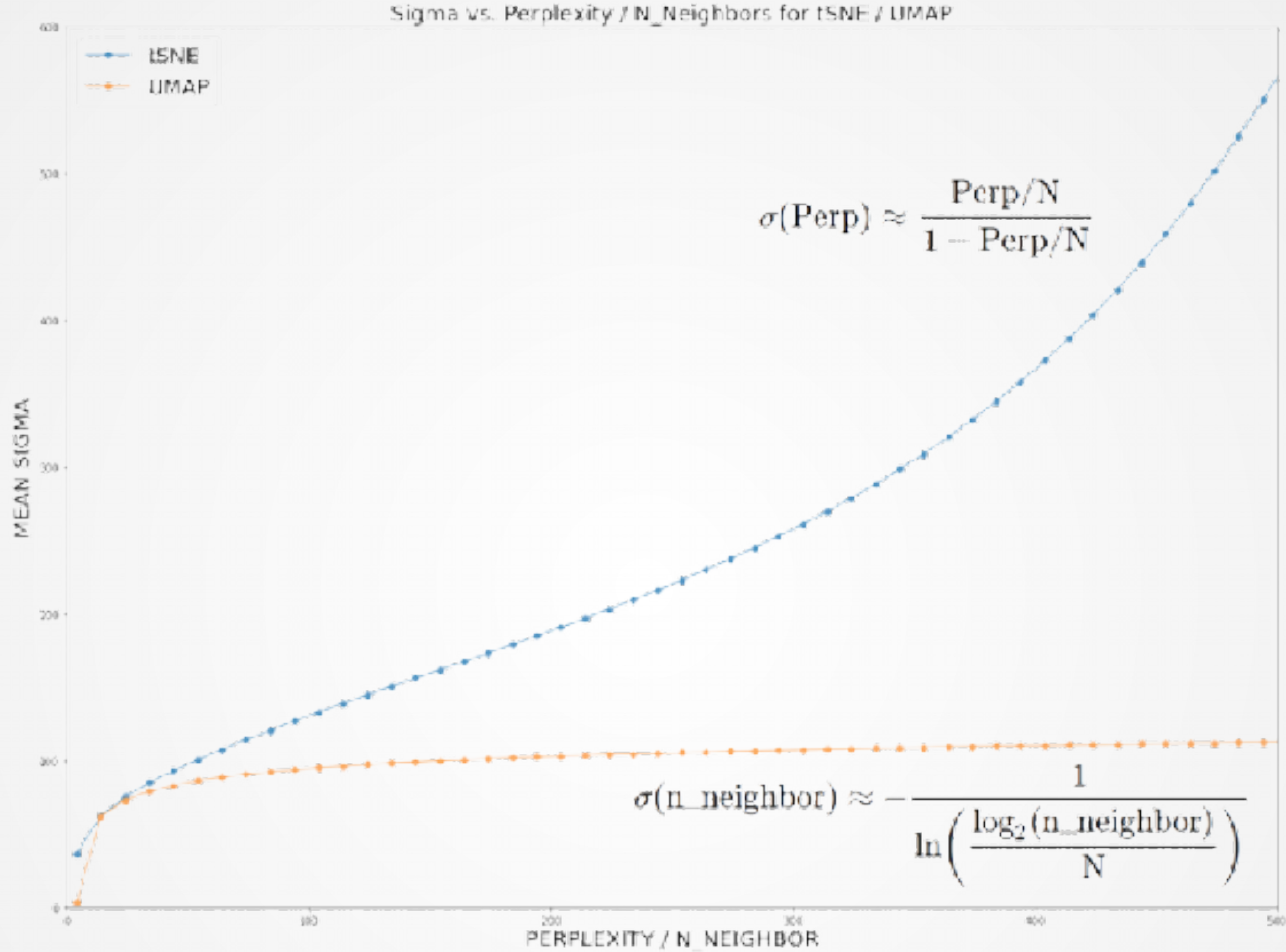
Cross Entropy (CE)

- Large X, get a huge CE penalty when Y is small



X: distance between points in high dimensions   Y: distance between points in low dimensions

Sensitivity towards hyperparameters



t-SNE is much more sensitive towards perplexity than UMAP towards n_neighbor

References

- http://www.sci.utah.edu/~beiwang/teaching/cs6965-fall-2019/
- https://towardsdatascience.com/dimensionality-reduction-for-data-visualization-pca-vs-tsne-vs-umap-be4aa7b1cb29
- http://www.sci.utah.edu/~beiwang/teaching/cs6965-fall-2019/Lecture02-DR.pdf
- https://nbisweden.github.io/excelerate-scRNAseq/saession-dim-reduction/lecture_dimensionality_reduction.pdf
- https://github.com/deniederhut/Slides-SciPyConf-2018/blob/master/umap/SciPy2018_UMAP.pdf
- https://speakerdeck.com/lmcinnes/umap-uniform-manifold-approximation-and-projection-for-dimension-reduction?slide=59

References

- http://www.sci.utah.edu/~beiwang/teaching/cs6965-fall-2019/
- https://towardsdatascience.com/dimensionality-reduction-for-data-visualization-pca-vs-tsne-vs-umap-be4aa7b1cb29
- http://www.sci.utah.edu/~beiwang/teaching/cs6965-fall-2019/Lecture02-DR.pdf
- https://nbisweden.github.io/excelerate-scRNAseq/saession-dim-reduction/lecture_dimensionality_reduction.pdf
- https://github.com/deniederhut/Slides-SciPyConf-2018/blob/master/umap/SciPy2018_UMAP.pdf
- https://speakerdeck.com/lmcinnes/umap-uniform-manifold-approximation-and-projection-for-dimension-reduction?slide=59

Dim Reduction VizTech

Elizaveta Zinovyeva
Lucas Umann
Bingling WANG
Wolfgang Karl Härdle

Ladislaus von Bortkiewicz Professor of Statistics
Humboldt-Universität zu Berlin
lvb.wiwi.hu-berlin.de