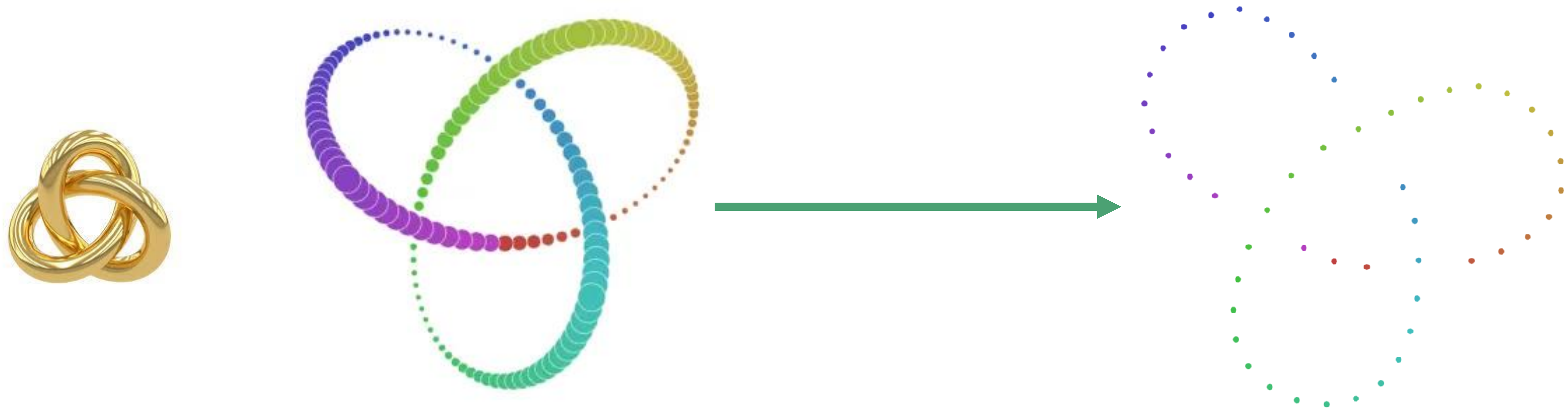
Visualizing High-Dimensional Data with tSNE & UMAP

Ethan, Evan, Electra, and Kara

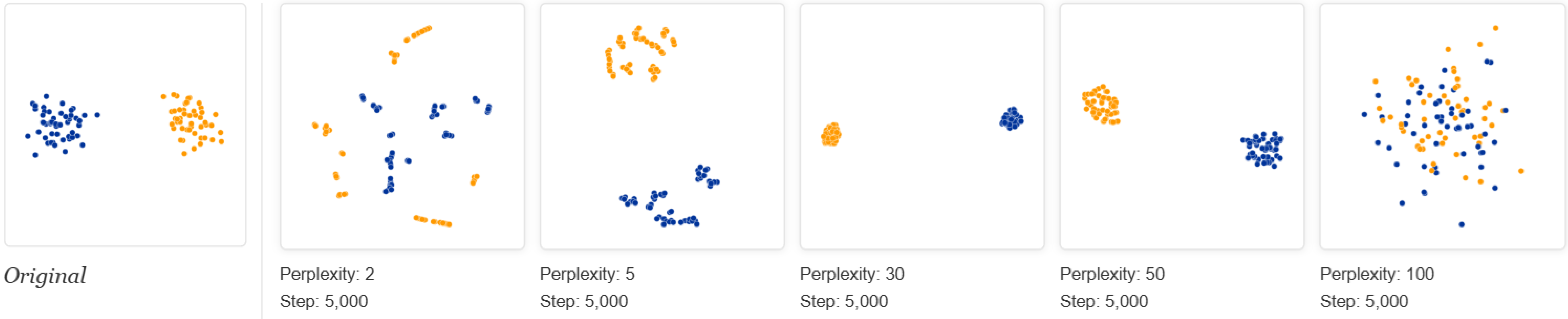
tSNE (t-distributed Stochastic Neighbor Embedding) - Introduction

- Like PCA, tSNE is a dimensionality reduction algorithm
- Used to visualize high-dimensional data by projecting it into a low dimensional space
- tSNE uses **nonlinear dimensionality reduction**
- It is an improvement on SNE and follows the same general algorithm process



tSNE Hyperparameters

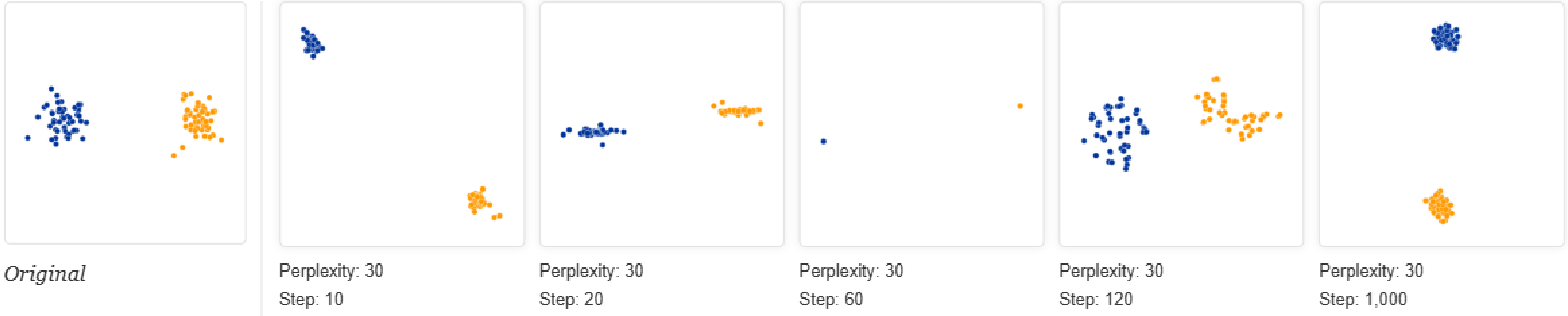
- Perplexity: Balance between local and global structure of data
  - Guesses # of close neighbors to a given point



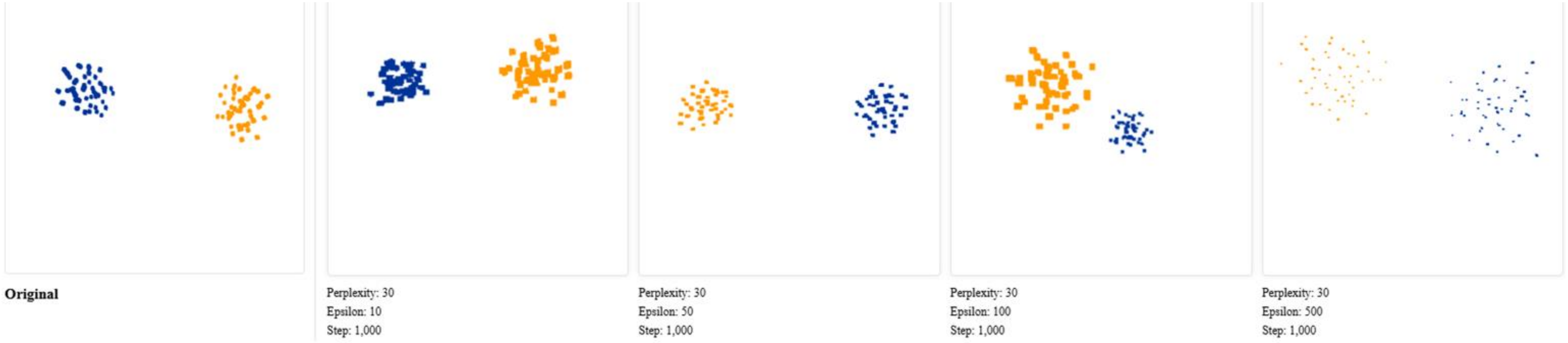
  - At low perplexity, local structure dominates
  - Perplexity should always be less than the number of points

tSNE Hyperparameters

- Iterations: # of steps to process data
  - Want enough iterations to reach stability

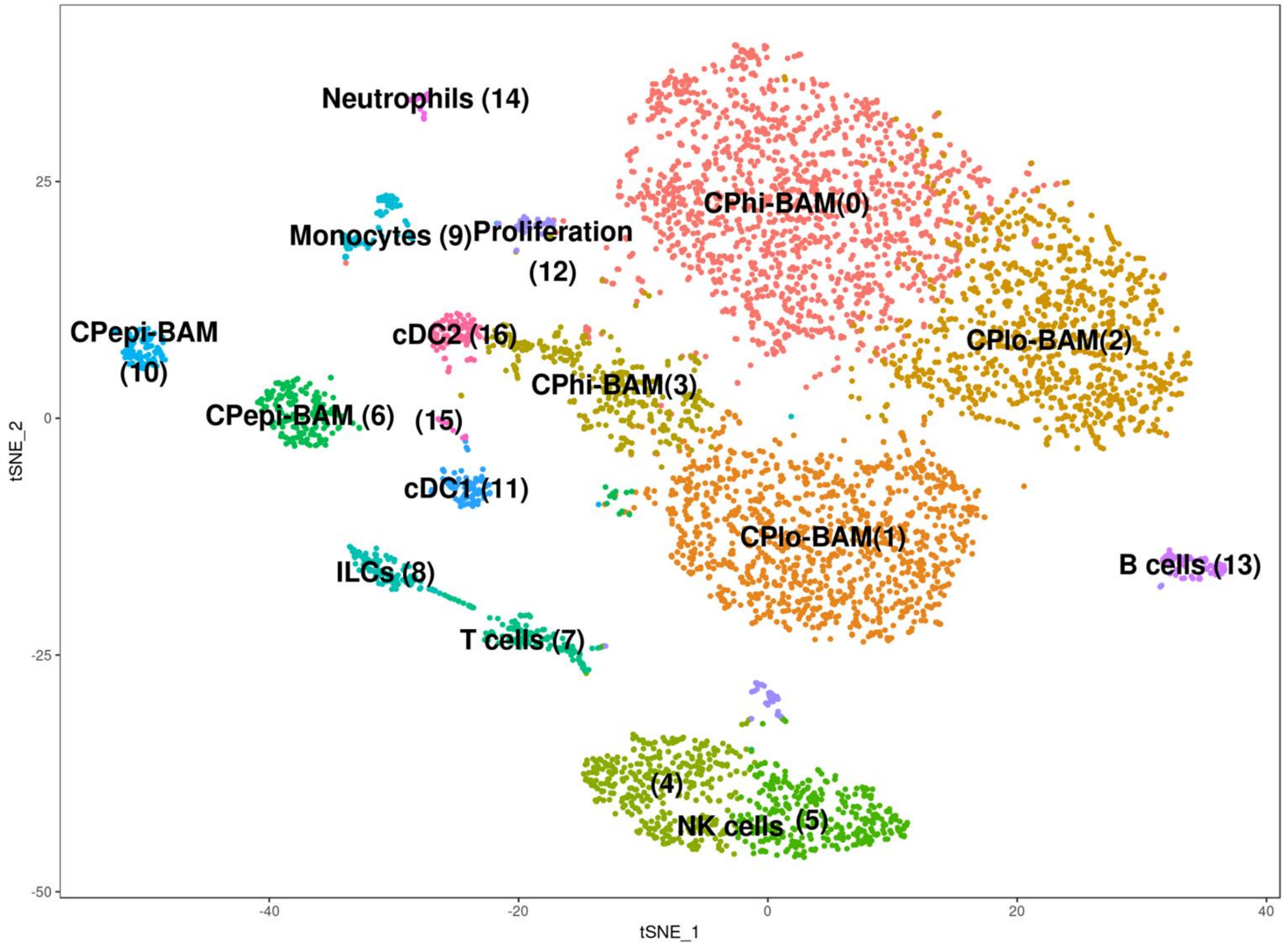


- Epsilon: Learning Rate
  - essentially controls the movement of the points in each step



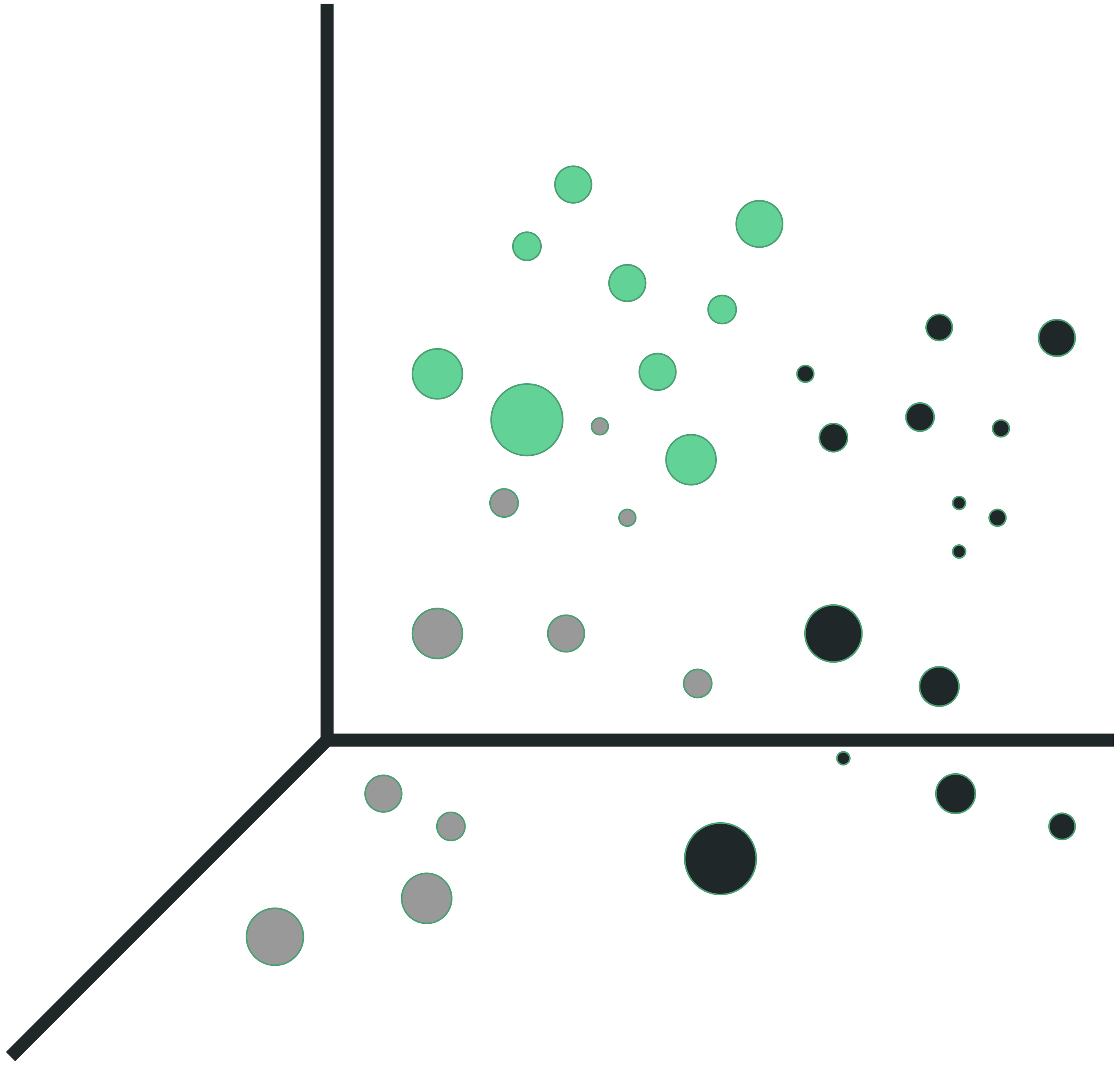
tSNE Graph Interpretation

- Every dot is a specific sample (ie, a cell)
- Cluster sizes mean nothing
- Cannot draw much from distances between clusters, unlike in PCA
  - tSNE does not place dots farther apart that are very different (poor global structure)
  - The axes are not directly interpretable
- You can make out shapes, generally, in the 2D plotted data
  - Patterns in a cluster can reveal information about that cluster



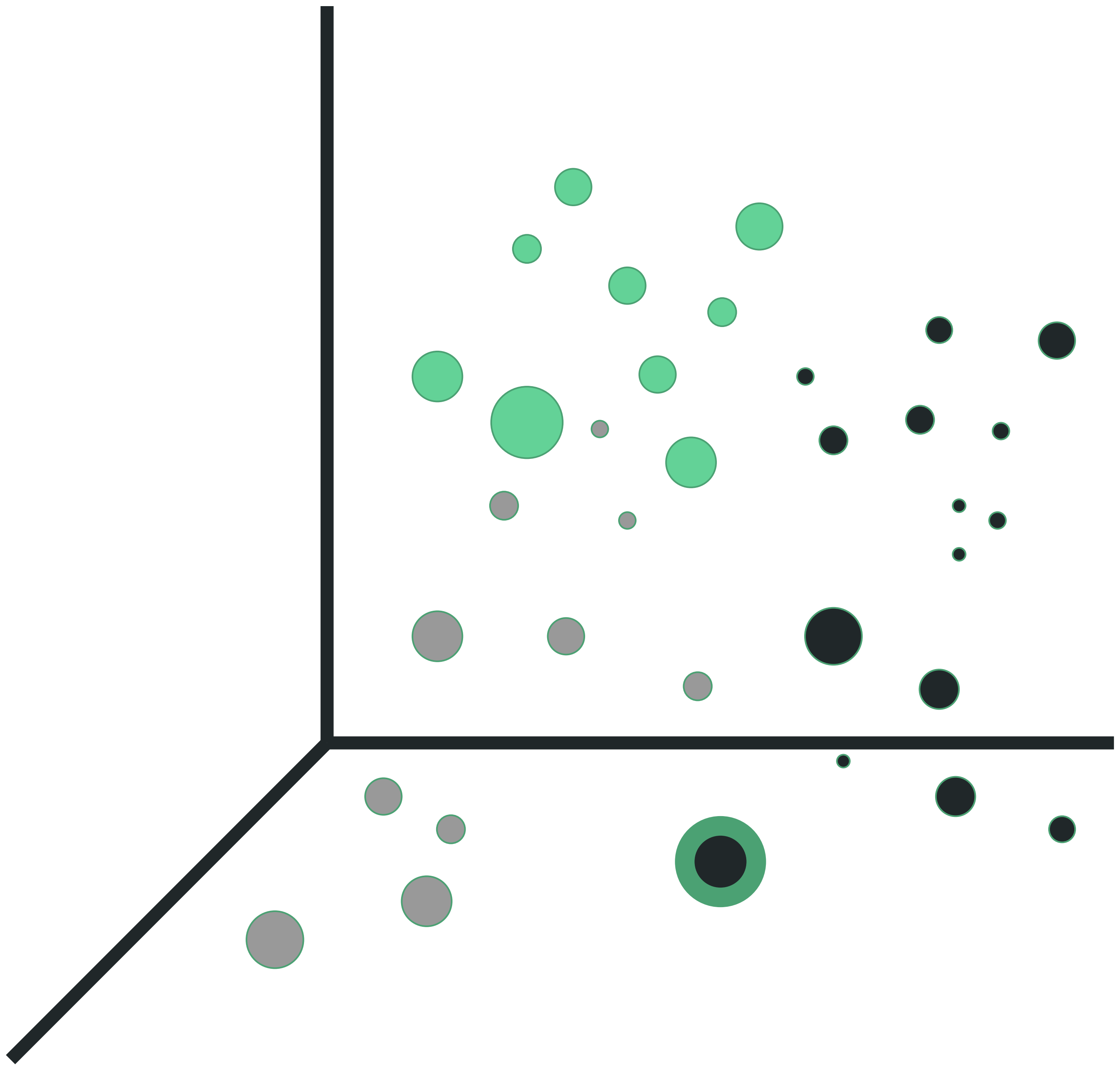
t-SNE: How it works

- Calculates the euclidean distance between points in higher dimensional space



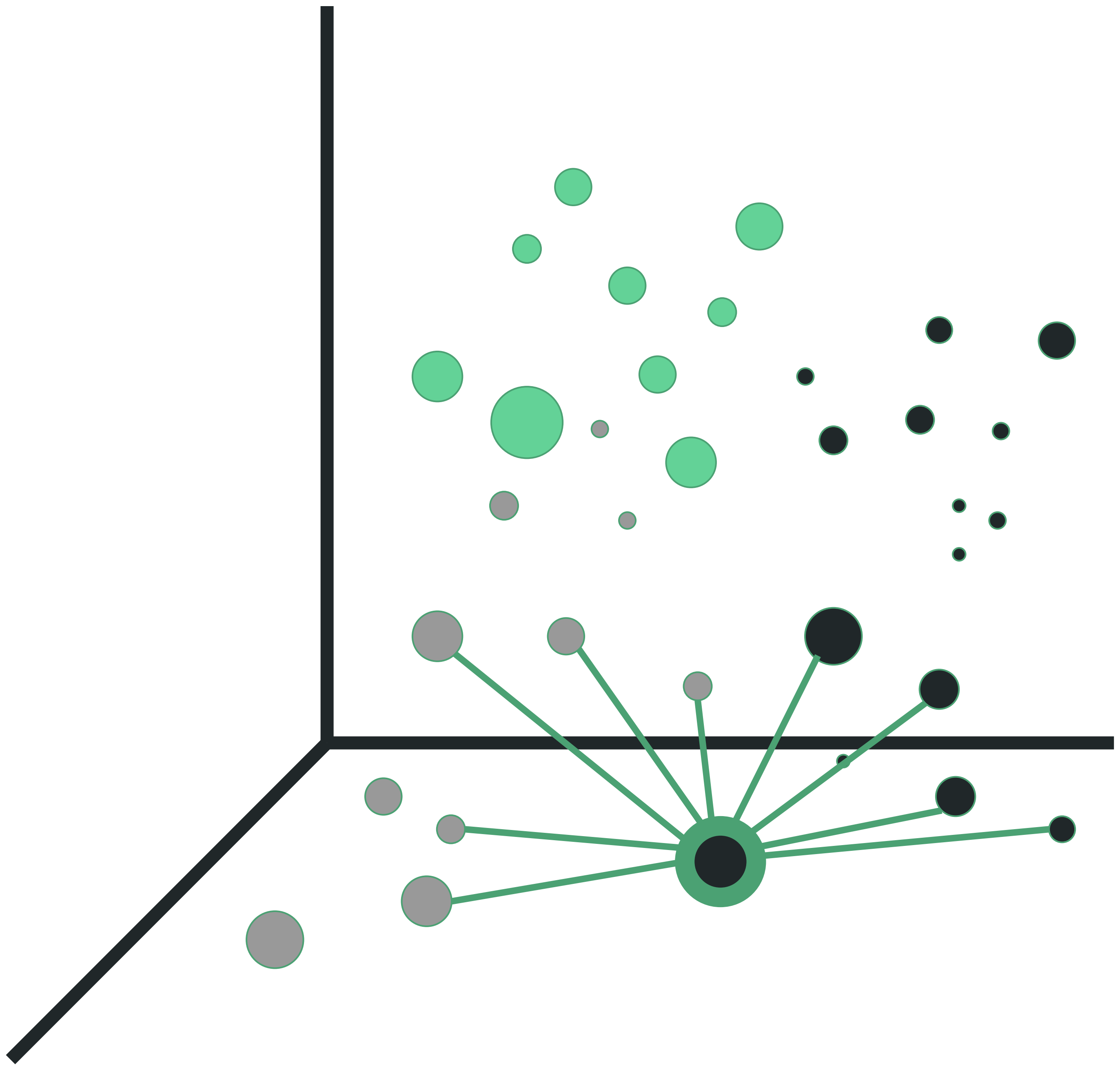
t-SNE: How it works

- Calculates the euclidean distance between points in higher dimensional space



t-SNE: How it works

- Calculates the euclidean distance between points in higher dimensional space



t-SNE: How it works

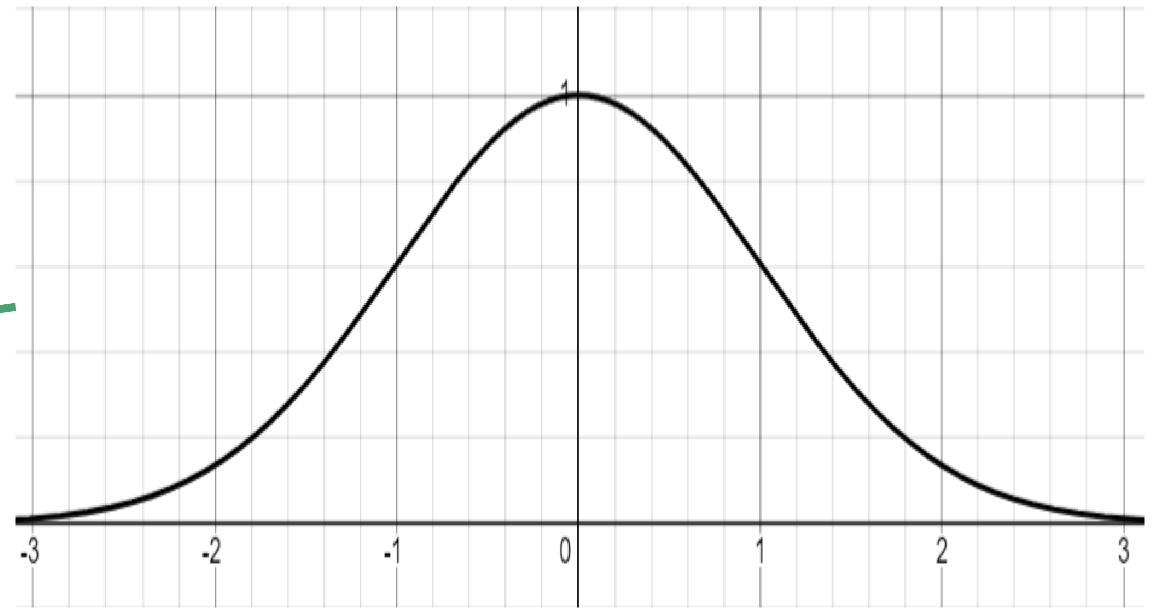
- Converts distances into probabilities using a Gaussian distribution

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

t-SNE: How it works

- Converts distances into probabilities using a Gaussian distribution

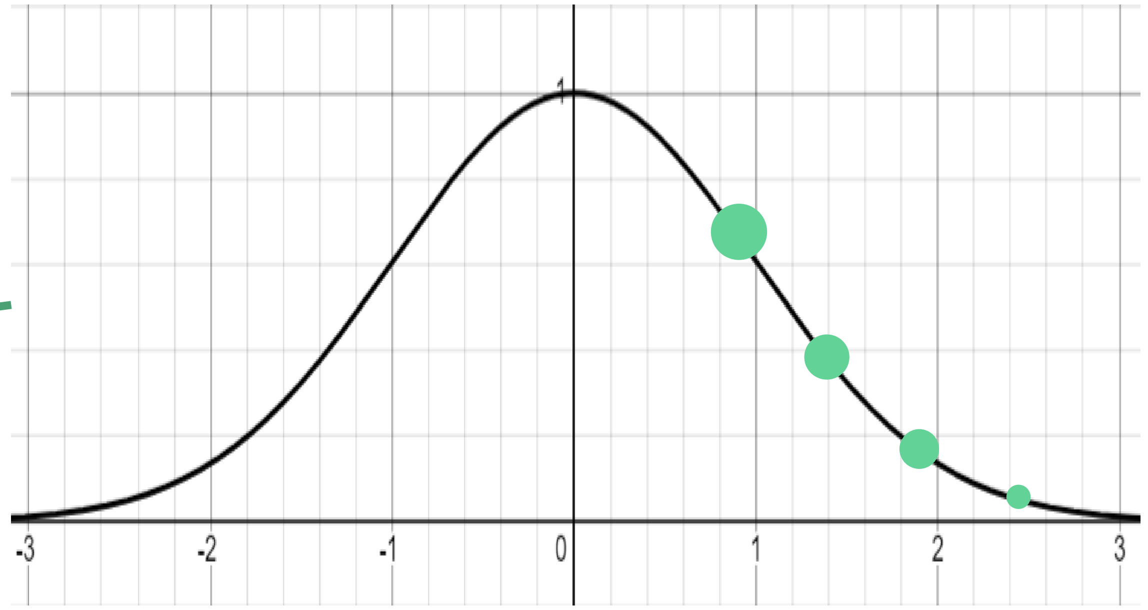
$$p_{j/i} = \frac{\exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(\frac{-\|x_i - x_k\|^2}{2\sigma_i^2}\right)}$$



t-SNE: How it works

- Converts distances into probabilities using a Gaussian distribution

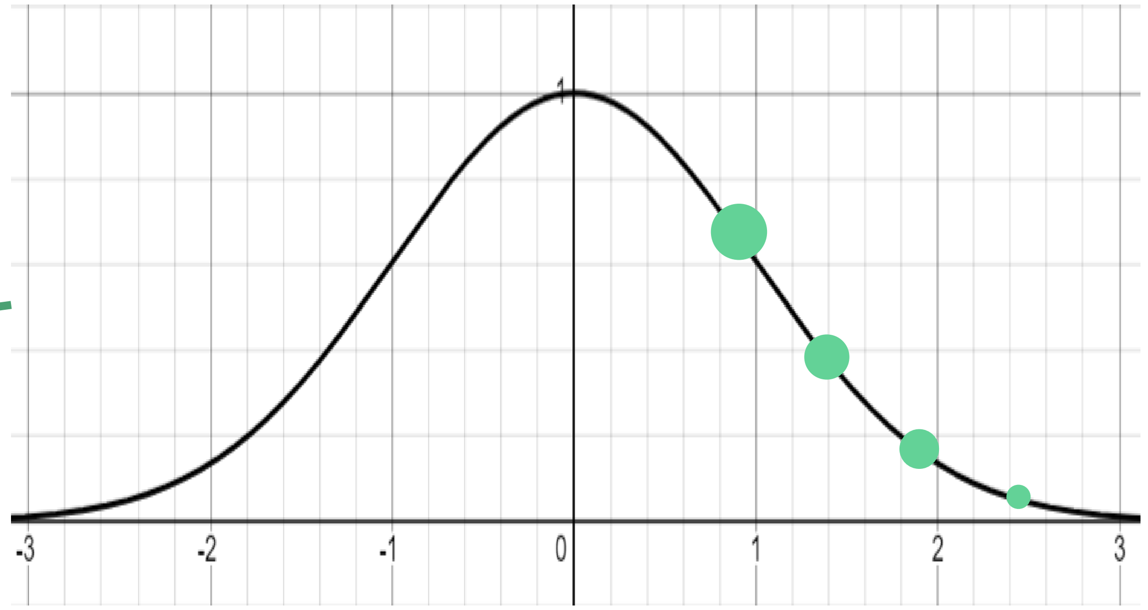
$$p_{j/i} = \frac{\exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(\frac{-\|x_i - x_k\|^2}{2\sigma_i^2}\right)}$$



t-SNE: How it works

- Converts distances into probabilities using a Gaussian distribution

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



sum of all distance probabilities

t-SNE: How it works

- Variance is set such that the entropy of the distribution is equal to the binary log of the user supplied perplexity
  - Perplexity essentially determines the largest distance the user considers neighbors

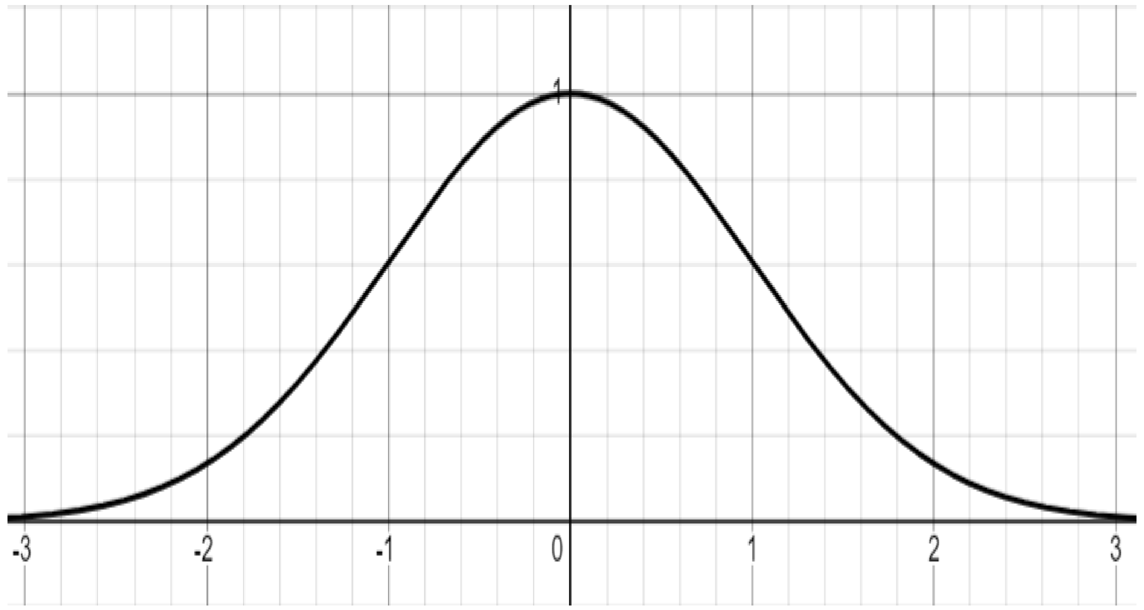
$$Perp(P_i) = 2^{H(P_i)}$$

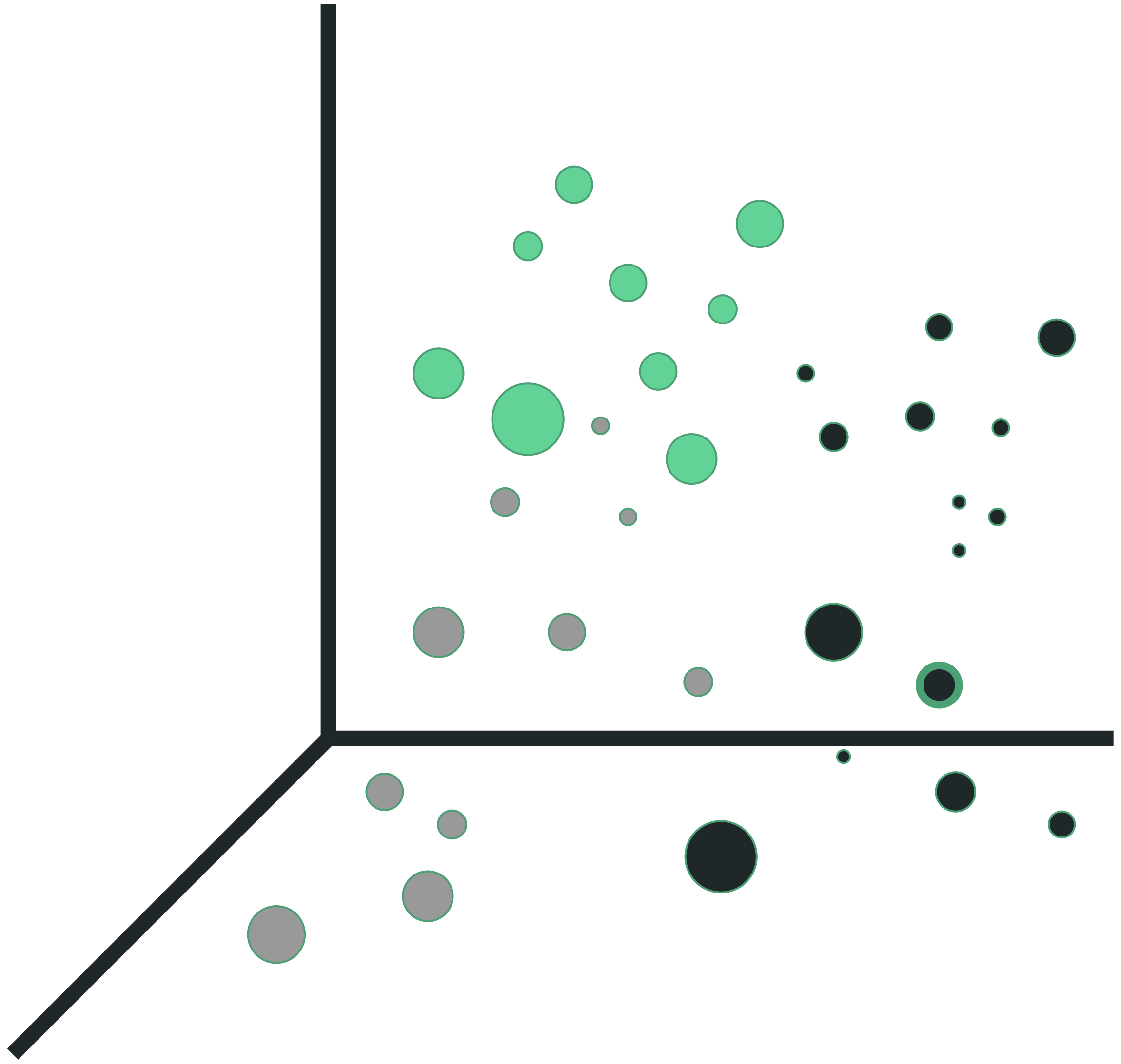
$$H(P_i) = -\sum_j p_{j|i} \log_2 p_{j|i}.$$

t-SNE: How it works

- Variance is set such that the entropy of the distribution is equal to the binary log of the user supplied perplexity
  - Perplexity essentially determines the largest distance the user considers neighbors

$\sigma^2 = 1$

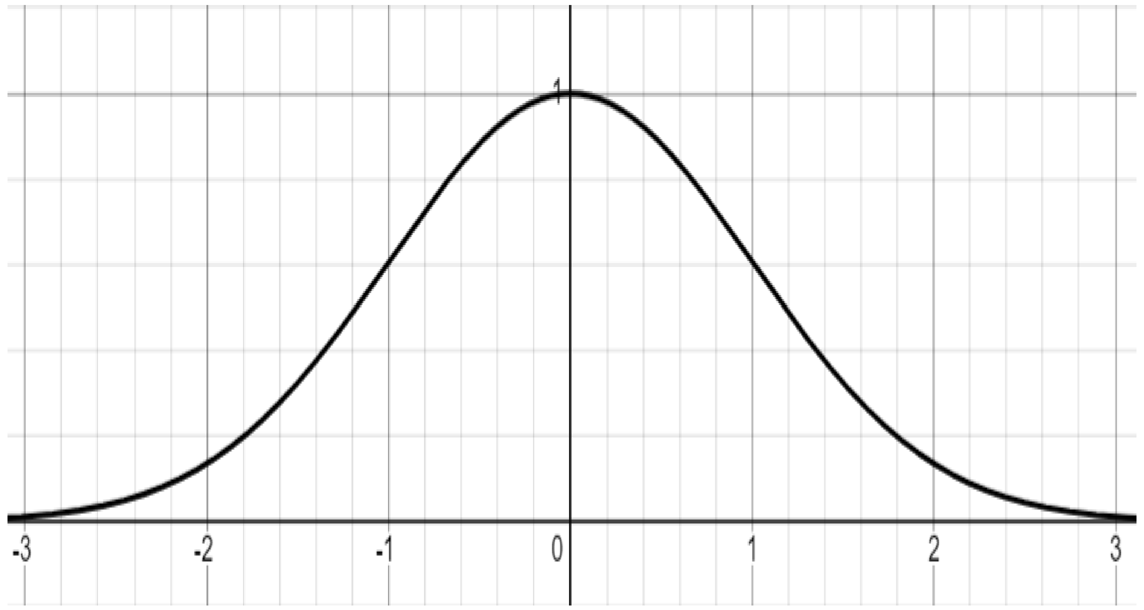


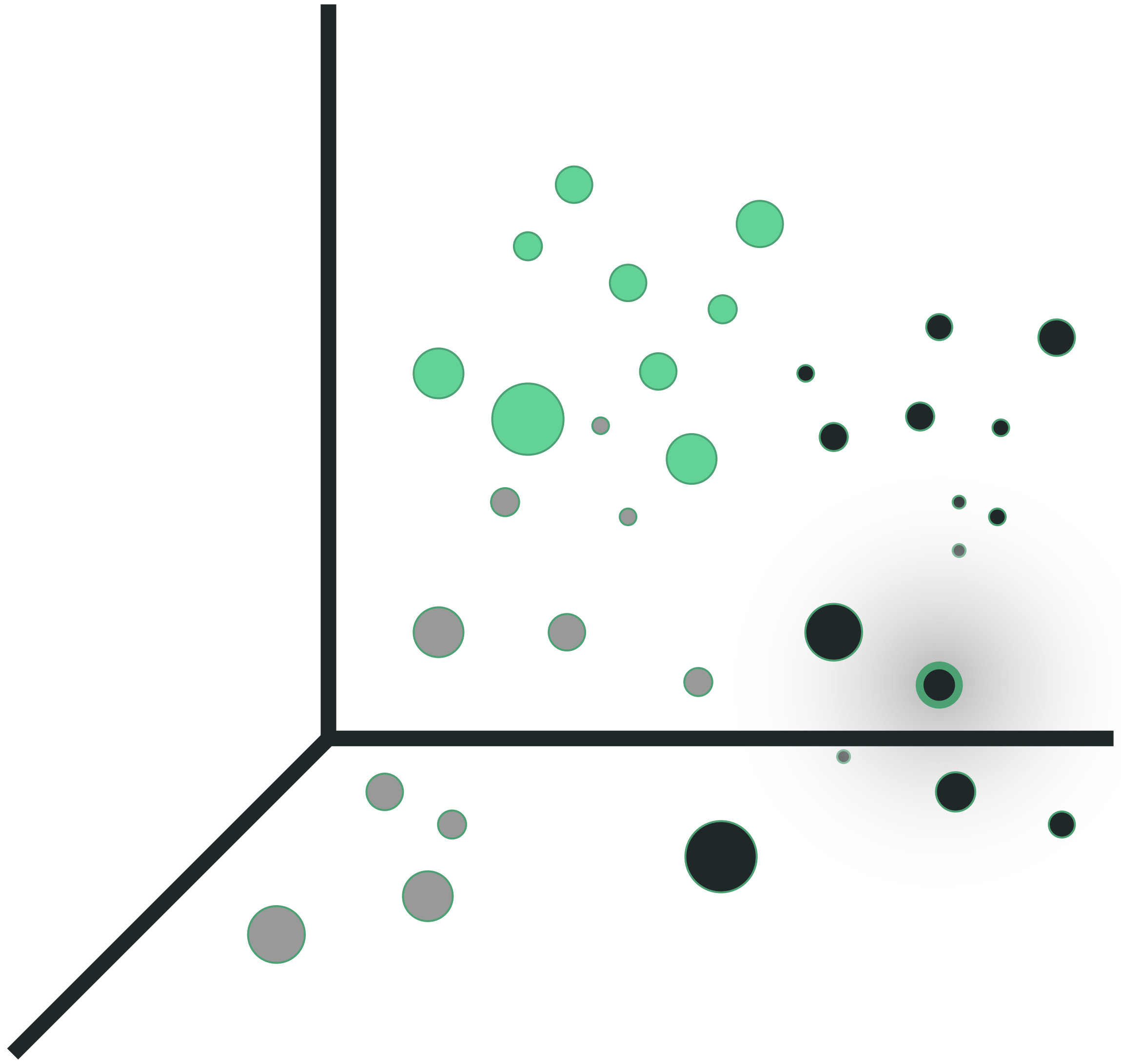


t-SNE: How it works

- Variance is set such that the entropy of the distribution is equal to the binary log of the user supplied perplexity
  - Perplexity essentially determines the largest distance the user considers neighbors

$\sigma^2 = 1$

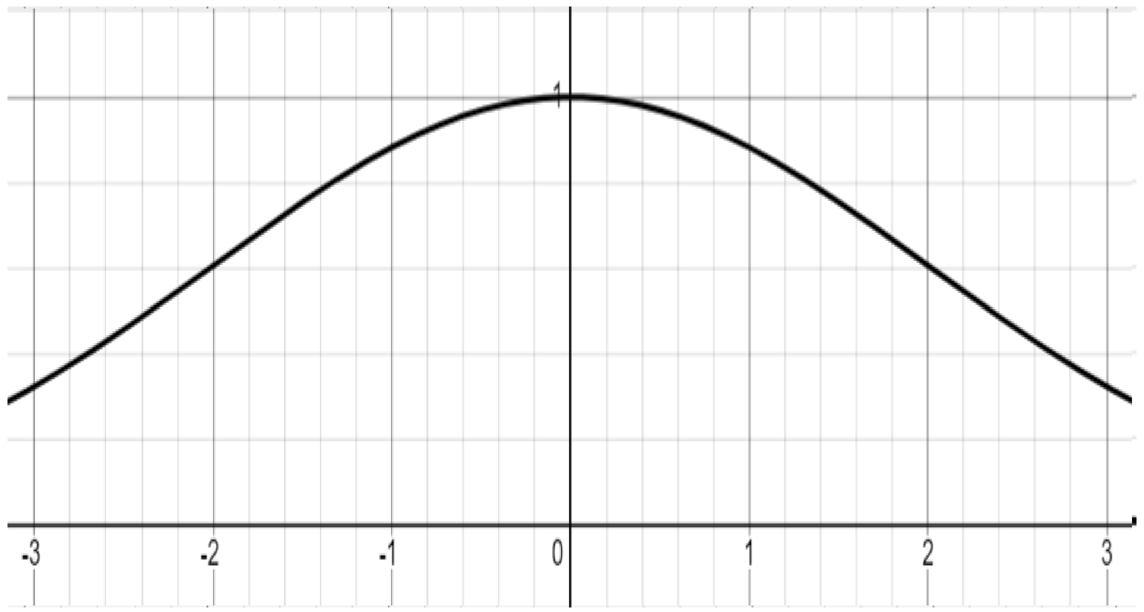


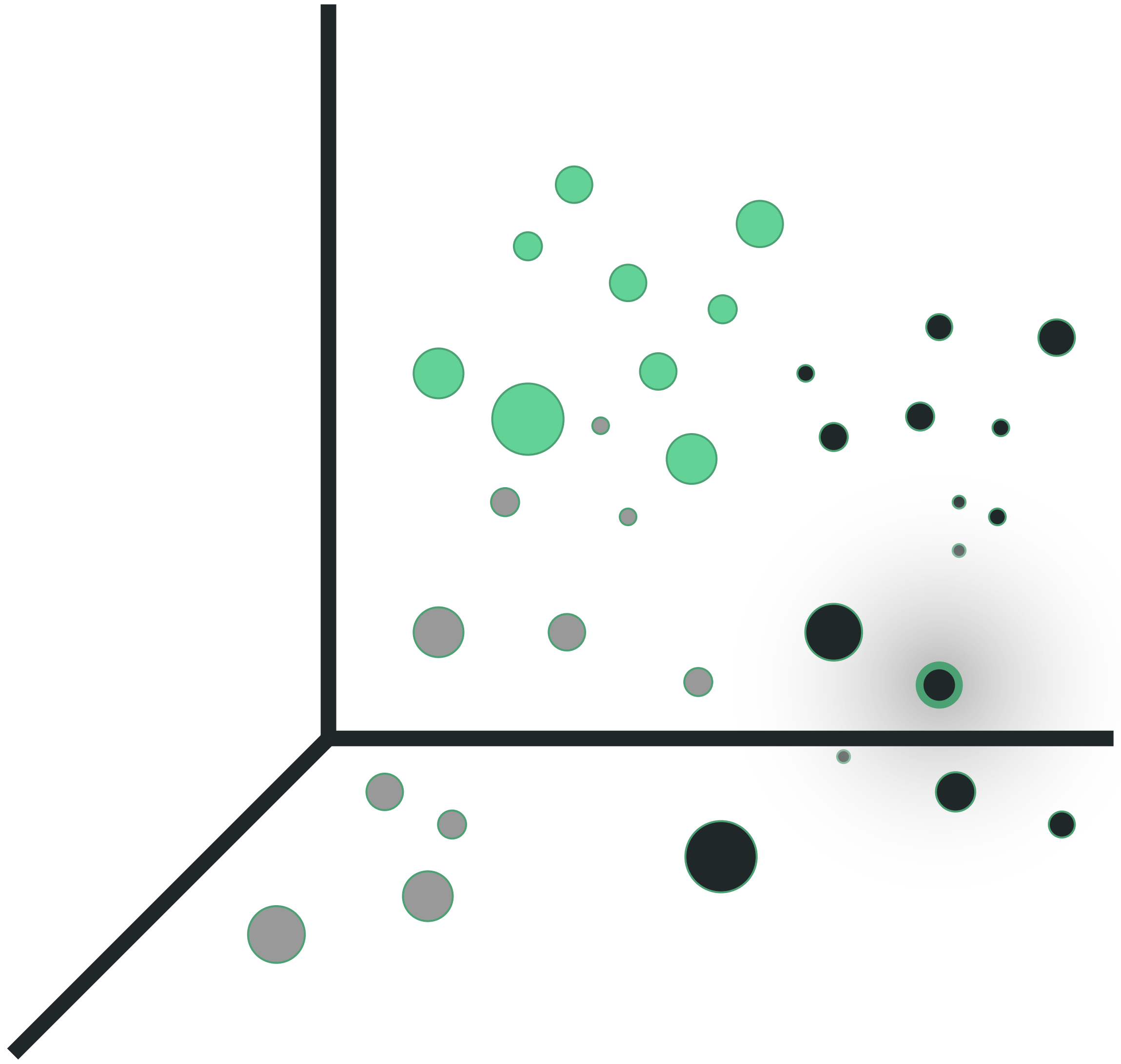


t-SNE: How it works

- Variance is set such that the entropy of the distribution is equal to the binary log of the user supplied perplexity
    - Perplexity essentially determines the largest distance the user considers neighbors

$\sigma^2 = 4$

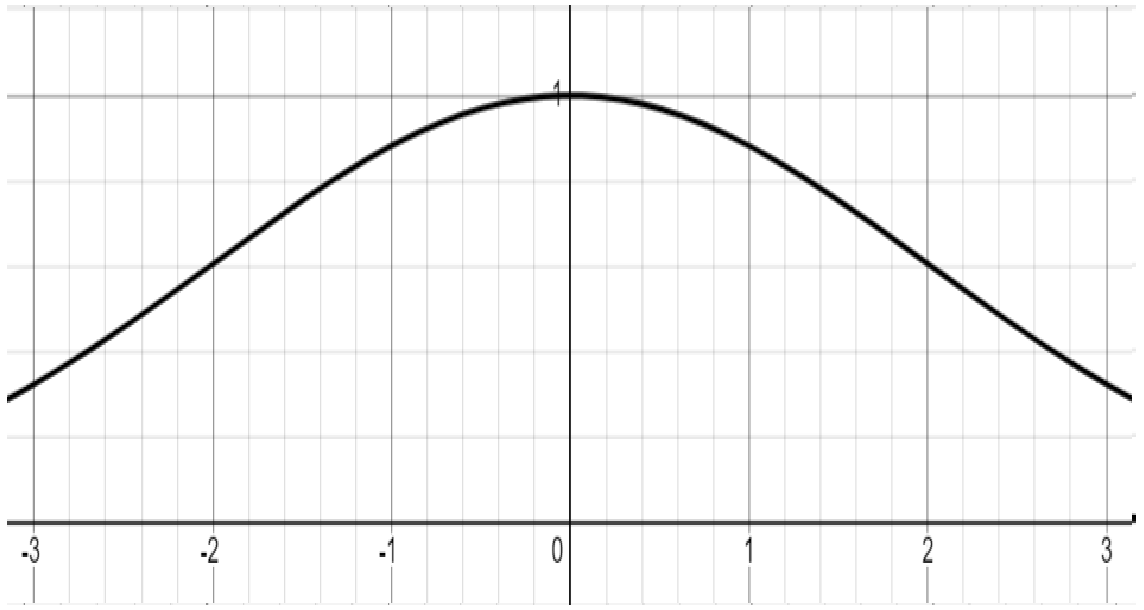


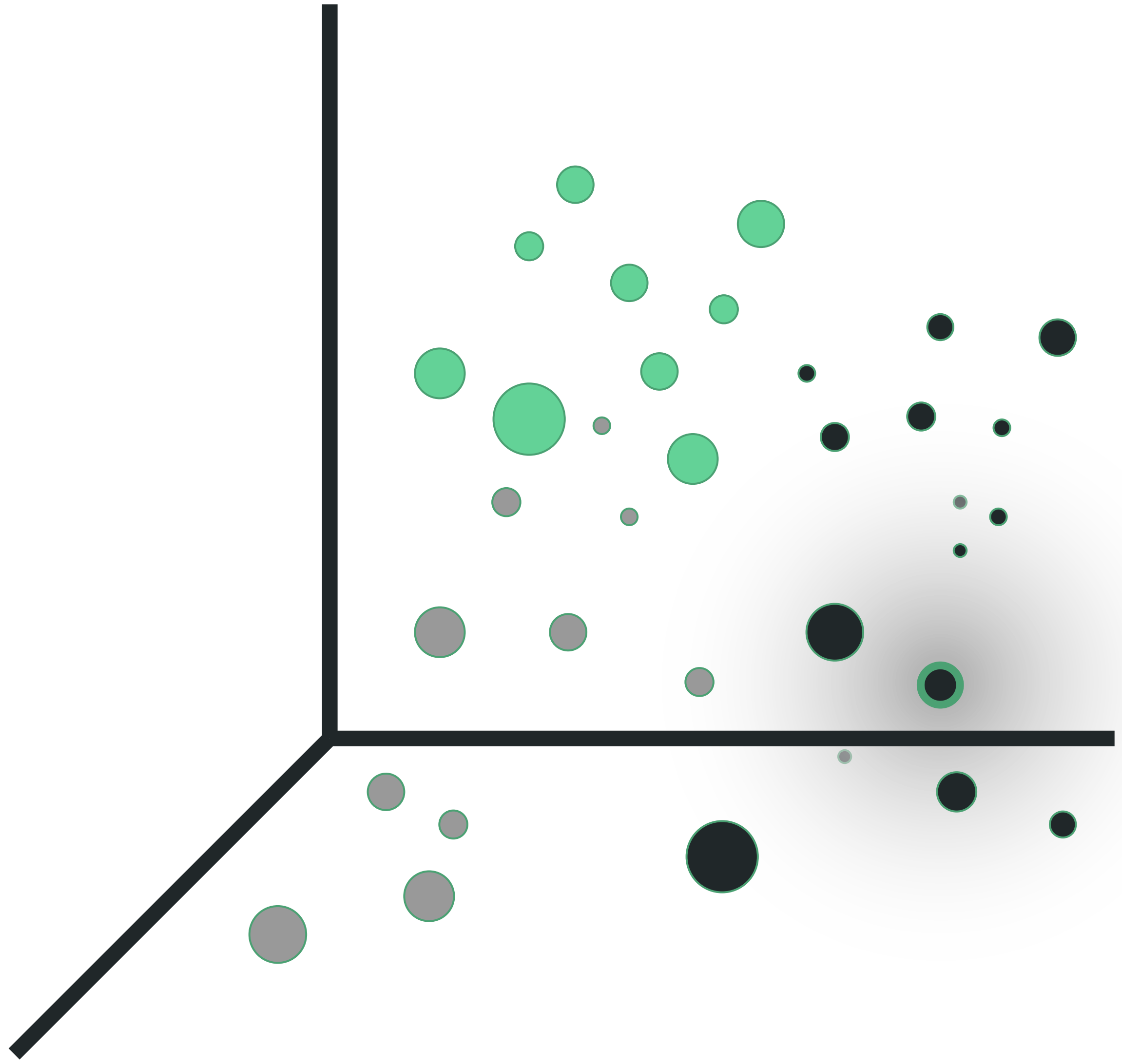


t-SNE: How it works

- Variance is set such that the entropy of the distribution is equal to the binary log of the user supplied perplexity
  - Perplexity essentially determines the largest distance the user considers neighbors

$\sigma^2 = 4$

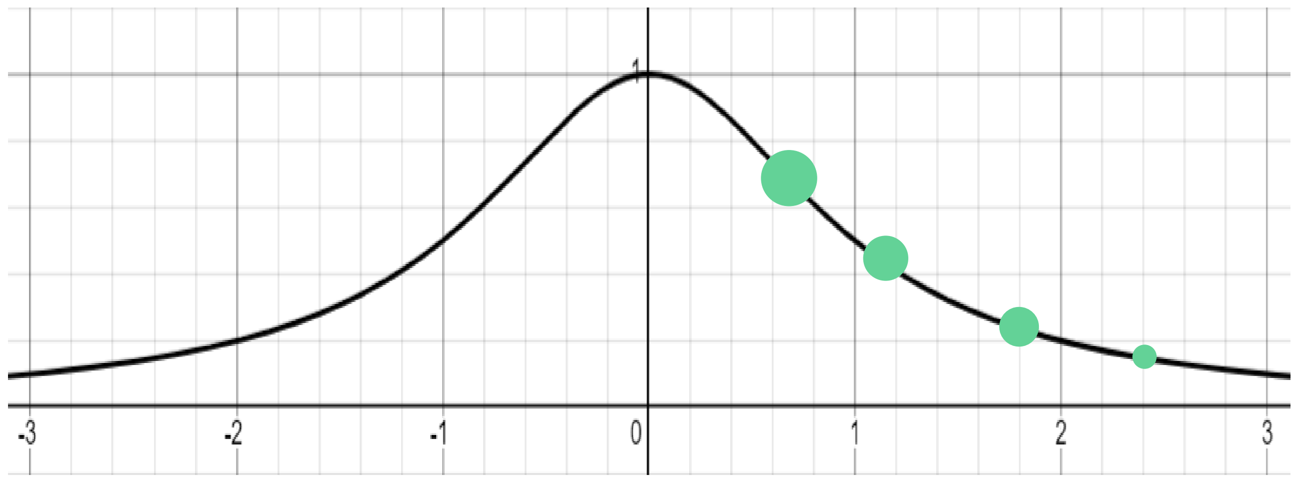




t-SNE: How it works

- Converts mapped distances into probabilities using a Student's t-distribution

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

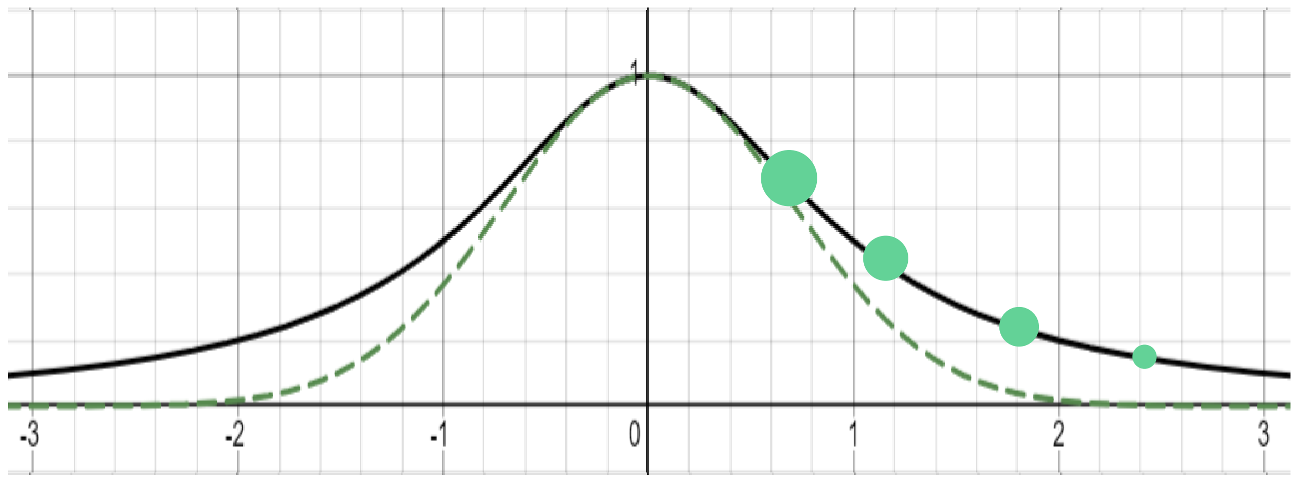


sum of all distance probabilities

t-SNE: How it works

- Converts mapped distances into probabilities using a Student's t-distribution

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



sum of all distance probabilities

t-SNE: How it works

- In low-dimensional space, the positions of the points are mapped so as to optimize the cost function
  - KL ➜ Kullback-Leibler Divergence

$$C = \sum_i KL\Big(P_i \mid |Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

t-SNE: How it works

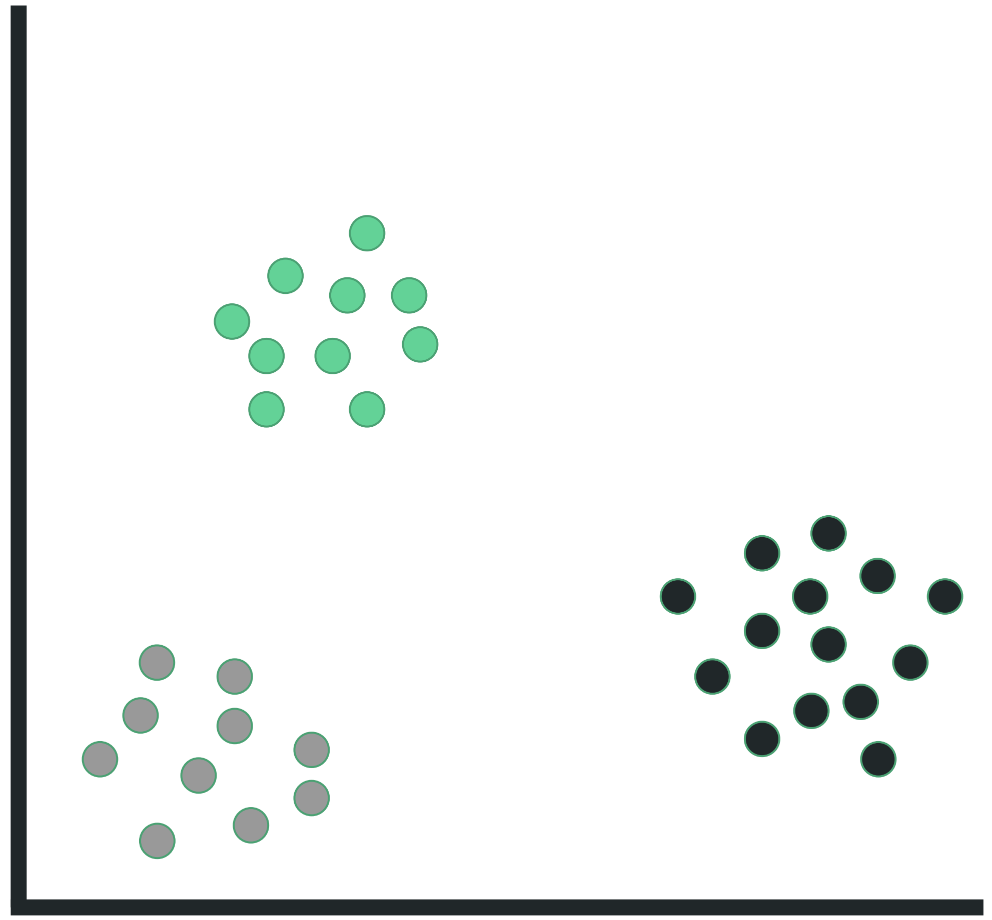
- This is done by matching the low-dimensional probability, $q_{j|i}$, to the high-dimensional probability

$$\frac{\exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(\frac{-\|x_i - x_k\|^2}{2\sigma_i^2}\right)} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}$$

t-SNE: How it works

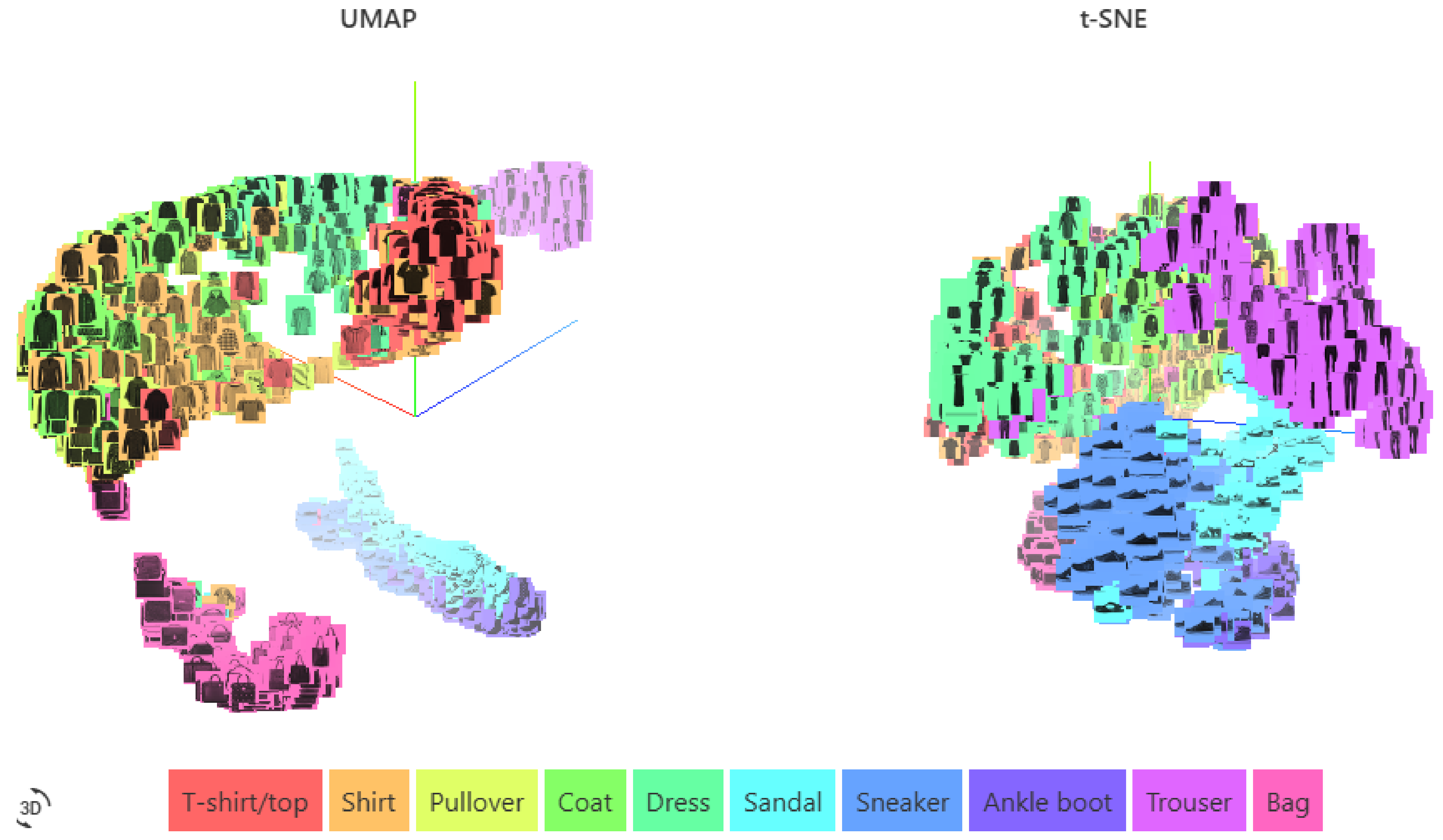
- This is done by matching the low-dimensional probability, $q_{j|i}$, to the high-dimensional probability

$$\frac{\exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma_i^2}\right)}{\sum_{k\neq i}\exp\left(\frac{-\|x_i - x_k\|^2}{2\sigma_i^2}\right)} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k\neq l}\left(1 + \|y_k - y_l\|^2\right)^{-1}}$$



Introduction to UMAP (Uniform Manifold Approximation and Projection)

- Also performs dimensionality reduction
- Increases speed and preserves the data's global structure
  - Algorithmic decisions justified by strong mathematical theory
  - spectral initialization of low dimensional graph
- UMAP more clearly shows similarities and differences between clusters



UMAP high-dimensional graph and Parameters

- UMAP builds a high-dimensional graph called a “fuzzy simplicial complex” before optimizing to a low-dimensional graph
- n_neighbors: number of approximate nearest neighbors used to construct the initial high-dimensional graph
  - Most important parameter, effectively how UMAP balances global vs local structure
- min_dist: minimum distance between points in low-dimensional space
  - Controls how tightly UMAP clumps points together

https://pair-code.github.io/understanding-umap/

Strengths and Weaknesses: t-SNE

STRENGTHS:

- Preserves local structure very well
- Good at revealing patterns in data
- Non-Linear

WEAKNESSES:

- Computationally intensive
- Cannot be used for preprocessing (more for visualization than dimensionality reduction)
- O(N*log(N))) complexity

Strengths and Weaknesses: UMAP

STRENGTHS:

- Runs very fast (usually faster than tSNE)
- More theoretical (mathematical) foundation than tSNE
- Good at preserving global structure
- Can handle larger datasets well (O(n))
- More scalable to large datasets

WEAKNESSES:

- High parameter sensitivity
- Non directly interpretable distances between points
- While more stable than tSNE, it is still stochastic
- Residual clusters may be defined that are artifacts of the algorithm
- Computationally intensive (< tSNE though)
- Less intuitive to newcomers than tSNE
- Non Convex Cost Functions

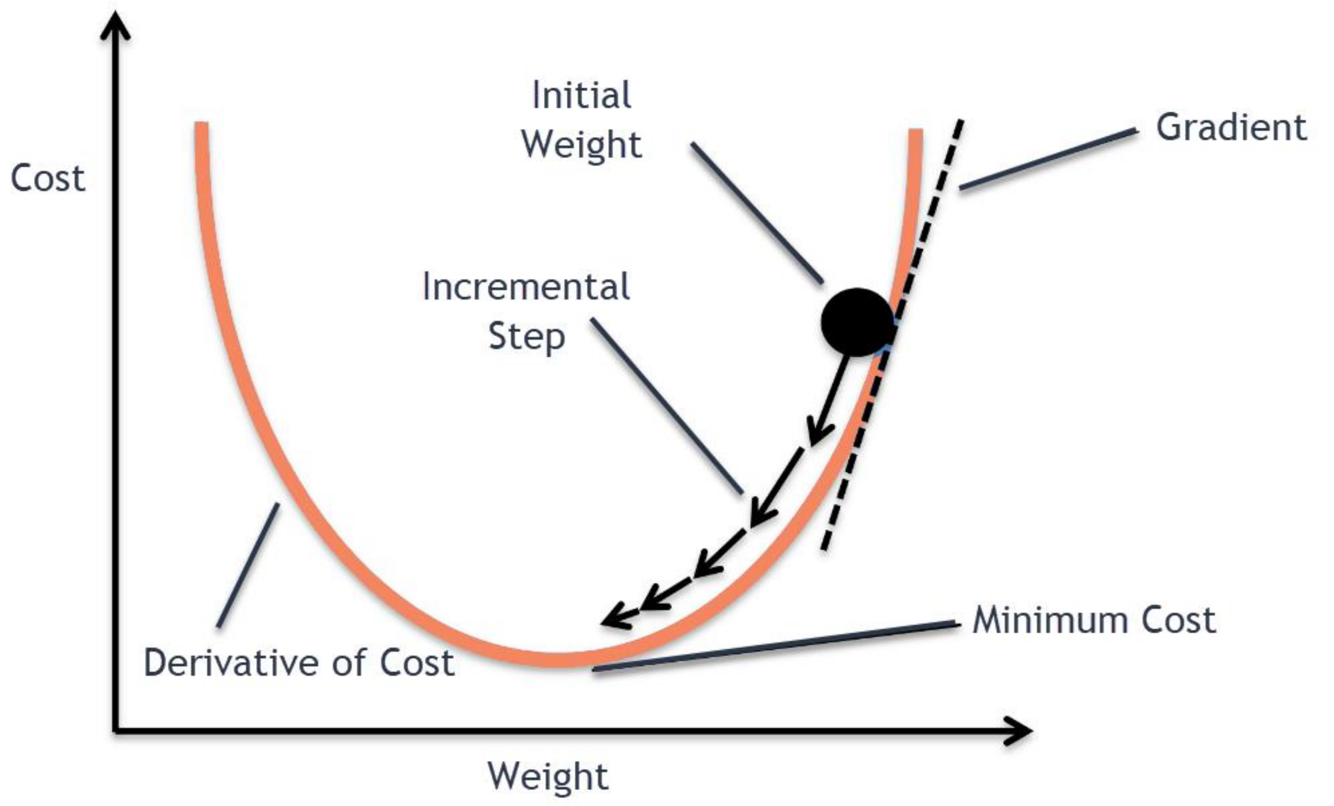
Strengths and Weaknesses: Cost Functions

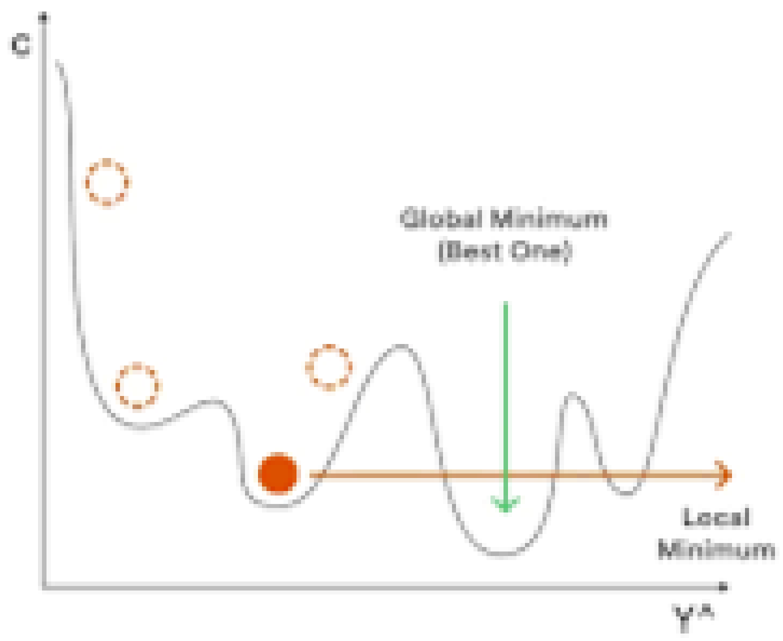
- UMAP uses Cross Entropy while tSNE uses KL-Divergence* (Oskolkov, 2019)
- This makes UMAP optimizable with stochastic gradient descent

$$H(P_{true}) + D_{KL}(P_{true}|P_{est}) = H(P_{true}, P_{est})$$

Entropy of P_{real}	D_{KL} between P_{real} and P_{est}	Cross-entropy between P_{real} and P_{est}
Optimal codeword length for P_{real}	Penalty for using P_{est} instead of P_{real}	Codeword length if optimal for P_{est} instead of P_{real}

Galagan, BE 562 Lecture Slides

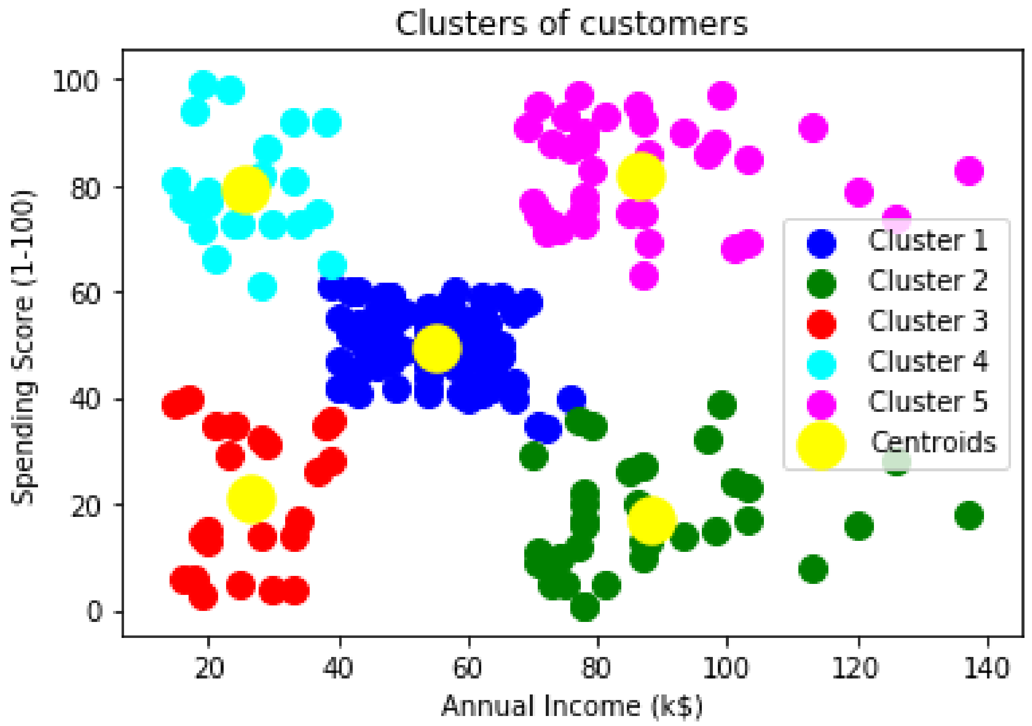




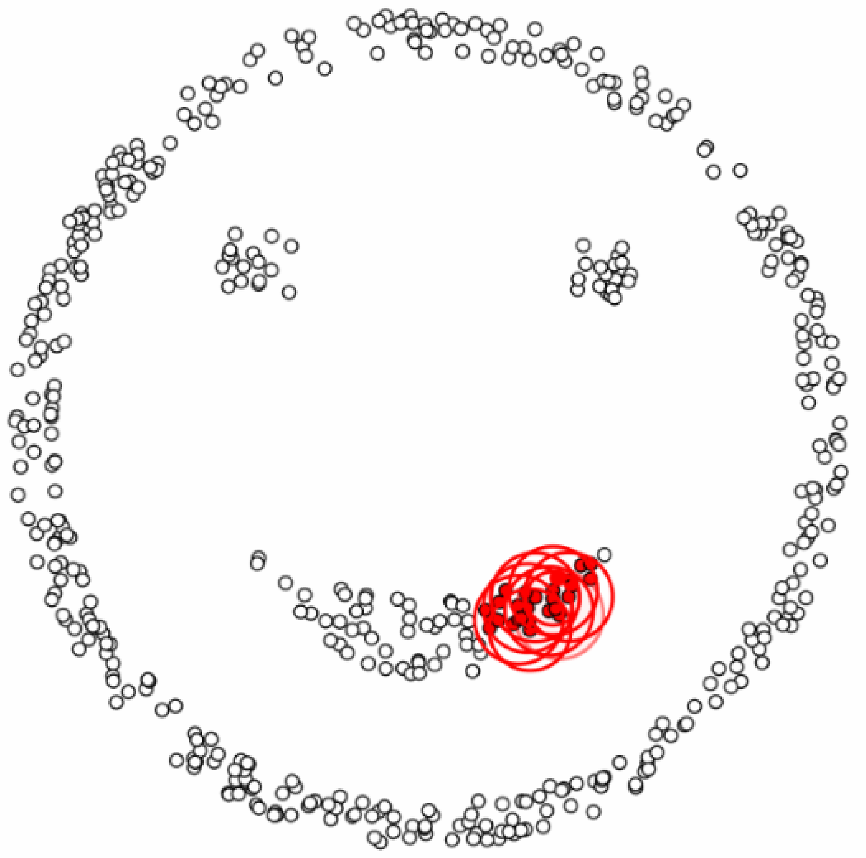
Strengths and Weaknesses: Summary

Metric	tSNE	UMAP
Global structure retention	ok	good
Local structure retention	great	good
Usability for non experts	ok	bad
Preprocessing?	no	yes
Computational intensity	bad (O(n*log(n))	still bad, but less (O(n))

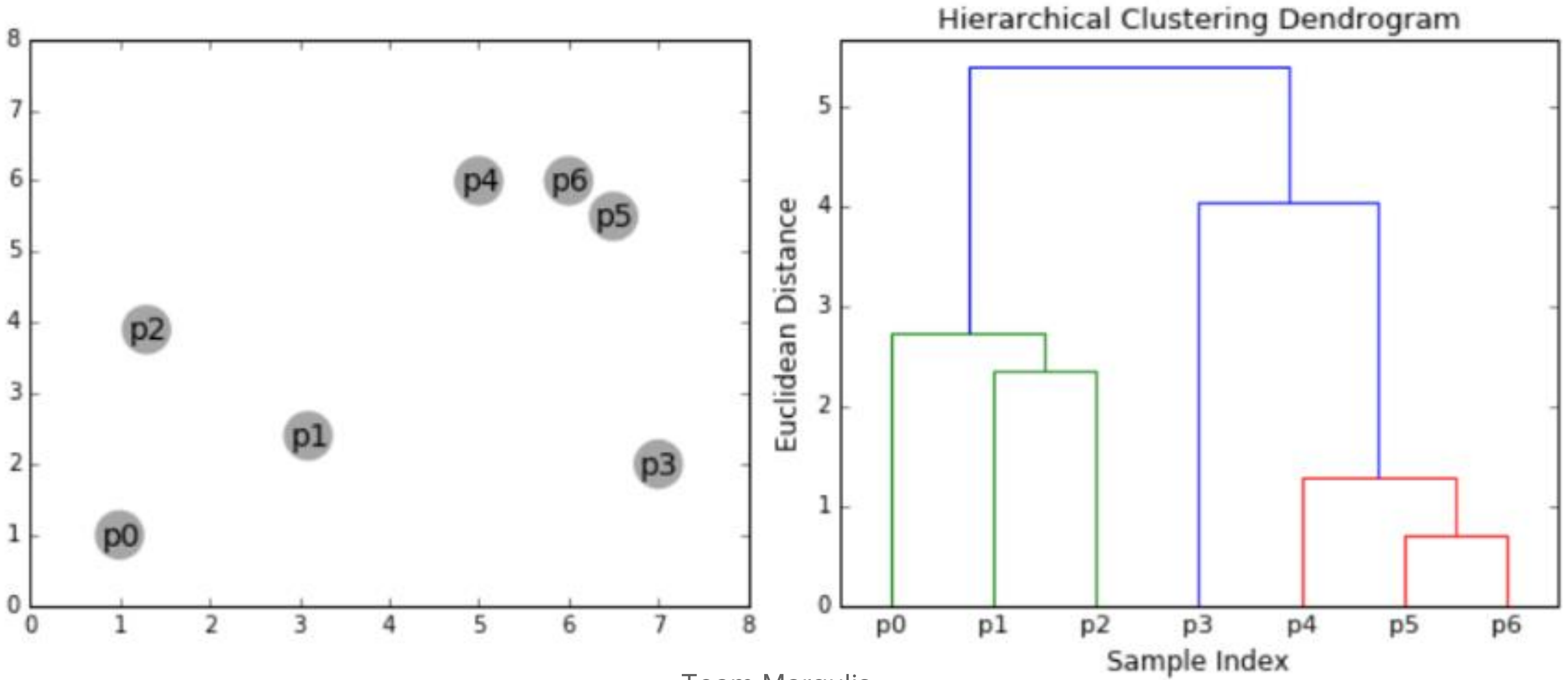
Refresher on Clustering Methods and PCA



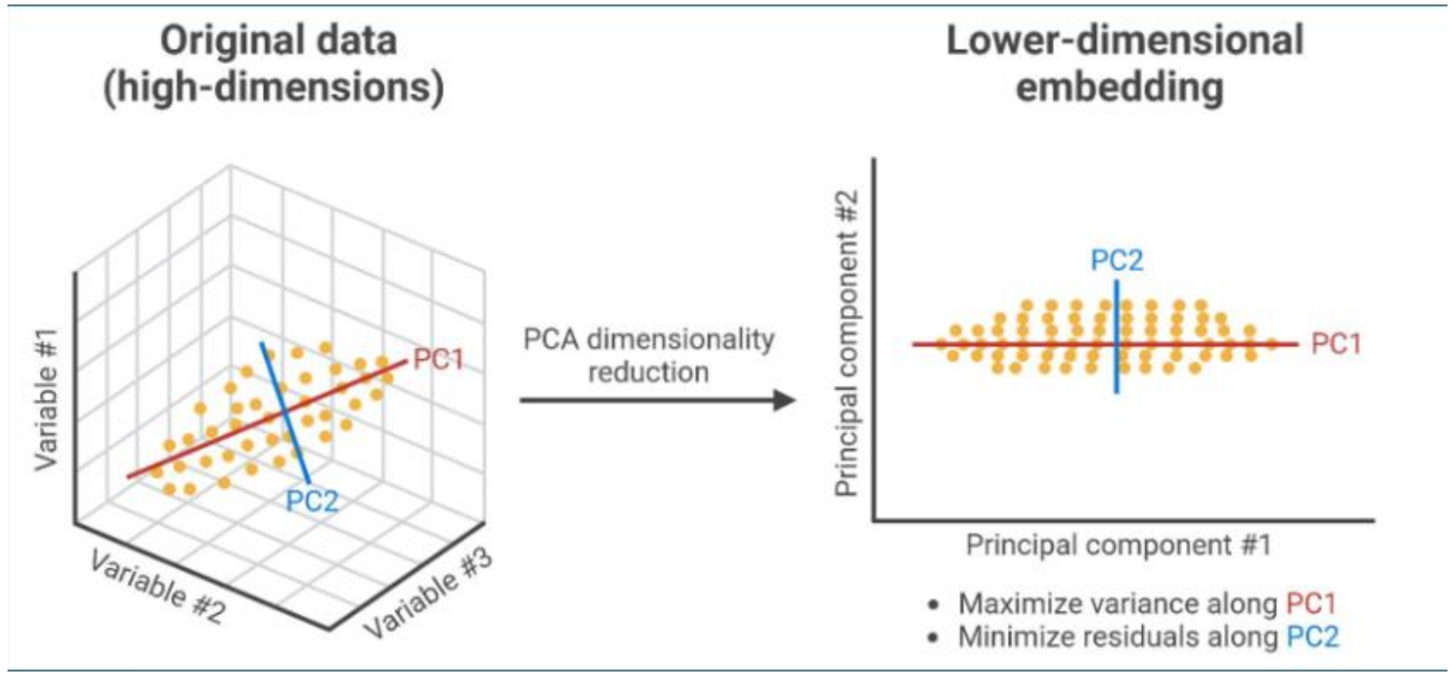
https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning



Team Margulis



Team Margulis



Team Brenner

Key Differences of t-SNE

	Clustering	PCA	t-SNE
Dimensionality reduction	no	for visualization and processing	for visualization
Applicable to a new dataset	parameters only	principle components	parameters only
Structure priority	N/A	global	local
Non-linearly separable data	depends	no	yes
Outlier handling	depends	poor	good
Computational strain	depends	relatively low	high for large dataset/high dim.

References

https://distill.pub/2016/misread-tsne/ (helpful for strengths/weaknesses)

https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a (good for tSNE maths)

https://www.geeksforgeeks.org/difference-between-pca-vs-t-sne/ (PCA vs tSNE)

https://wedadanbtawi95.github.io/tsne/#:~:text=The%20second%20parameter%20in%20t,the%20algorithm%20with%20random%20values.

https://e-archivo.uc3m.es/rest/api/core/bitstreams/ff0eaee9-3736-4854-9529-ac3c45d058ff/content

https://www.scdiscoveries.com/blog/knowledge/how-to-interpret-a-t-sne-plot/
https://medium.com/data-folks-indonesia/the-underlying-idea-of-t-sne-6ce4cff4f7

https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html

vandermaaten08a.pdf

https://pair-code.github.io/understanding-umap/

https://www.brainimmuneatlas.org/tsne-cp-irf8.php

t-Distributed Stochastic Neighbor Embedding | SpringerLink

https://towardsdatascience.com/how-exactly-umap-works-13e3040e1668

Kullback–Leibler divergence - Wikipedia