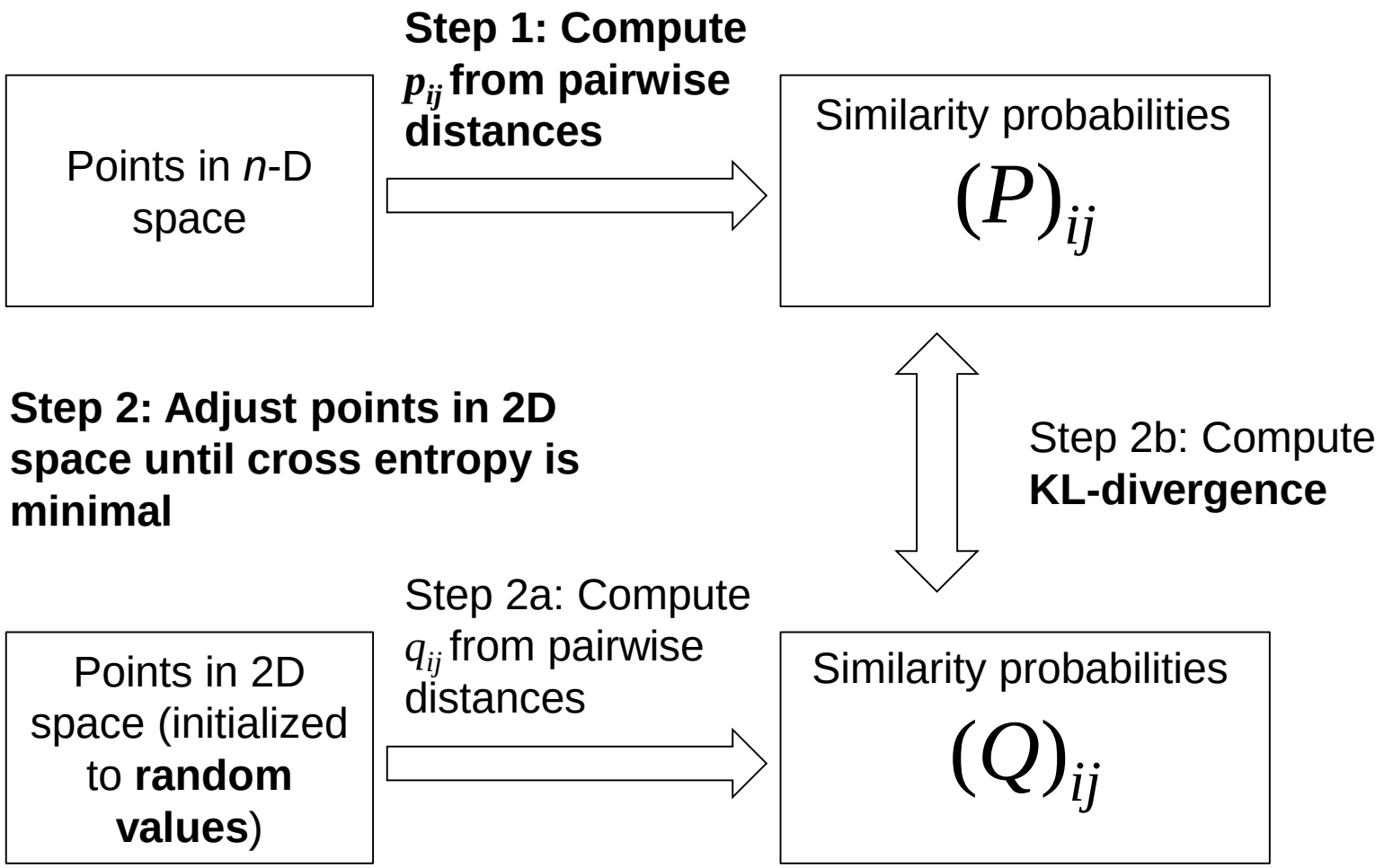
Dimensionality Reduction Part 4: t-SNE and UMAP

Ng Yen Kaow

Dimensionality Reduction

- Linear methods
  - PCA (Principal Component Analysis)
  - cMDS (Classical Multidimensional Scaling)
- Non-linear methods
  - KPCA (Kernel PCA)
  - mMDS (Metric MDS)
  - Isomap
  - LLE (Locally Linear Embedding)
  - **t-SNE** (t-distributed Stochastic Neighbor Embedding)
  - **UMAP** (Uniform Manifold Approximation and Projection)

t-SNE in a nutshell

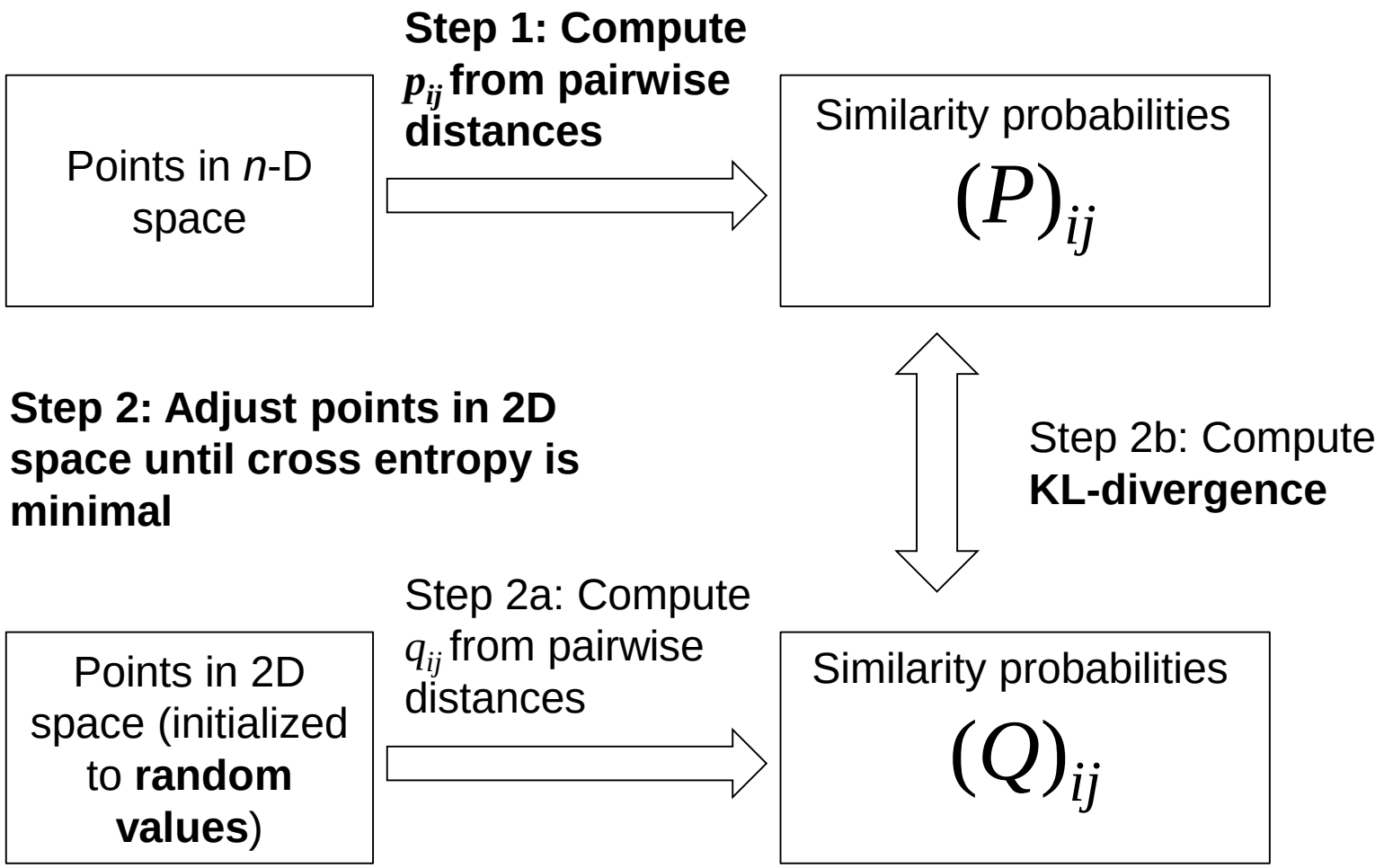


Step 2c: Modify points in 2D by **gradient descent**

t-SNE in a nutshell

- Transform *n*-D space points into 2D
- Convert the distances in *n*-D space between pairwise points i, j into their **similarity probability** p_{ij}, assuming
  - Gaussian probability fall-off by the distance of the pair
    - Each point is at the mean of its own Gaussian with its own variance (determined through “perplexity”)
  - Result is matrix $(P)_{ij}$
- Reconstruct distances in 2D space (graph layout) such that they give rise to a similarity probability matrix $(Q)_{ij}$ with
  - t-distribution probability fall-off by the distance of each pair in the reconstructed points
    - The t-distributions all have the same variance (making clusters evenly sized in the 2D space for better visualization)
  - Minimal KL-divergence from $(P)_{ij}$

t-SNE in a nutshell



Step 2c: Modify points in 2D by **gradient descent**

t-SNE details

- How to compute probability p_{ij}
  - $p_{ij} = \left(p_{j|i} + p_{i|j}\right)/2N, p_{ii} = 0$

Gaussian

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

Normalization

where the variance σ_i is found (through binary search) to fulfill

$$-\sum_j p_{j|i}\log_2 p_{j|i} = \log_2(\text{Perplexity})$$

for a user-defined hyperparameter Perplexity

- Gaussian used because the fast drop-off gives more weight to nearer points

t-SNE details

- How to compute probability q_{ij}

t-distribution

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

- t-distribution used because of the slower drop-off (or fatter tails)

- For the gradient descent watch this video

  https://www.youtube.com/watch?v=W-9L6v_rFIE&t=244s

- Code example

  https://github.com/karpathy/tsnejs

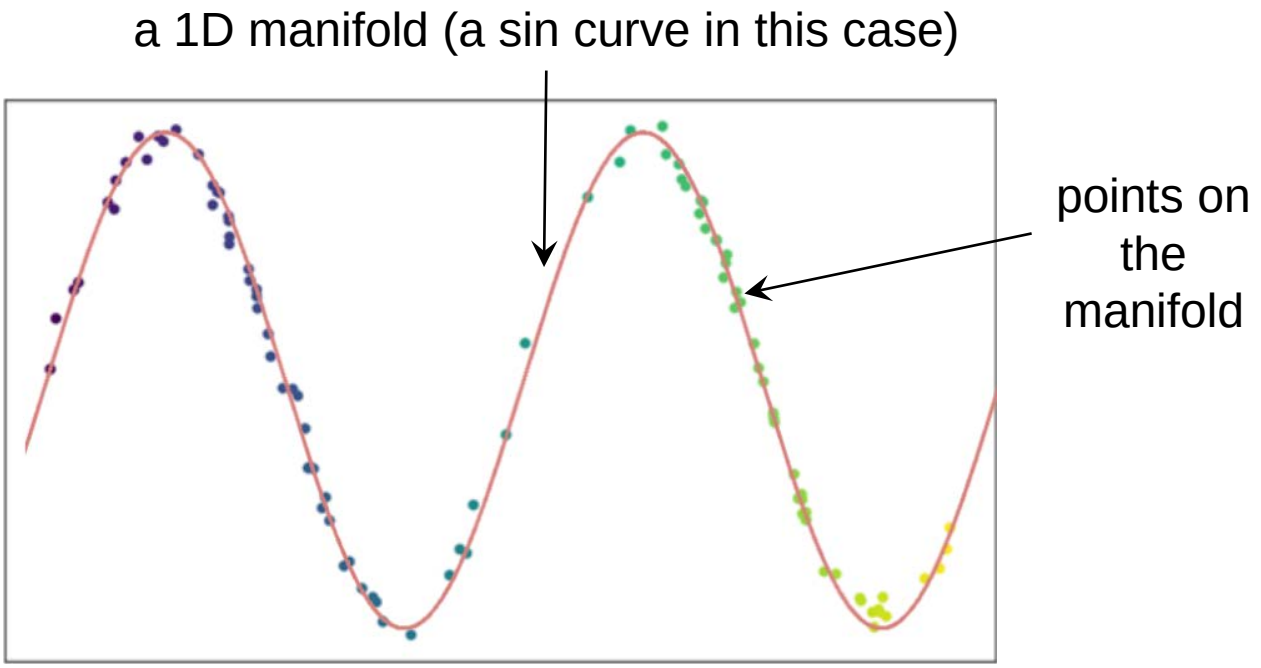
BTW, the most viewed t-SNE video on YouTube turns out to be very bad!

t-SNE lesson learned

- $(P)_{ij}$ is a good candidate for a distance matrix
  - Natural elimination of edges between distant points
  - Distance is adjusted for local data density through $\mathrm{Perplexity}$
    - However the justification for $\mathrm{Perplexity}$ is weak
    - This is solved in UMAP
- $(Q)_{ij}$ may be good for visualization, but may not be suitable as a distance matrix

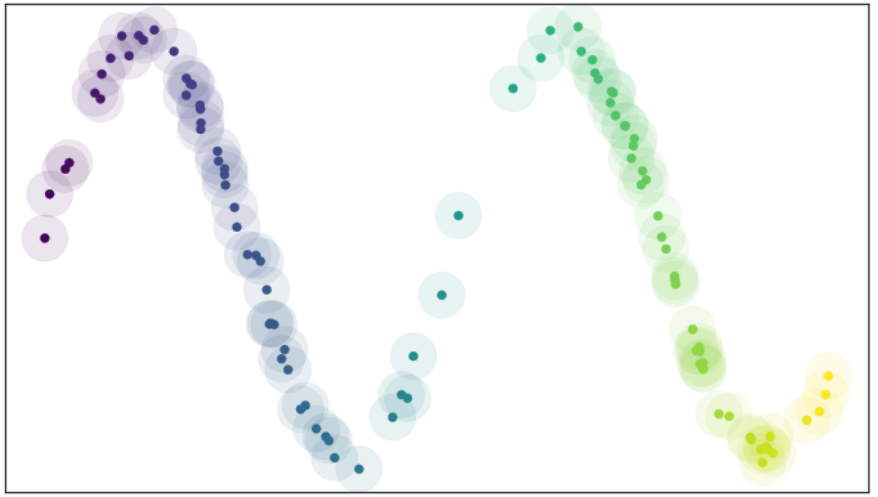
UMAP idea

- t-SNE converts the distance between two points into a probability in an *ad hoc* manner and attempts to preserve this probability
- UMAP starts with a totally different idea: To construct a manifold from only a sampling of the points on it

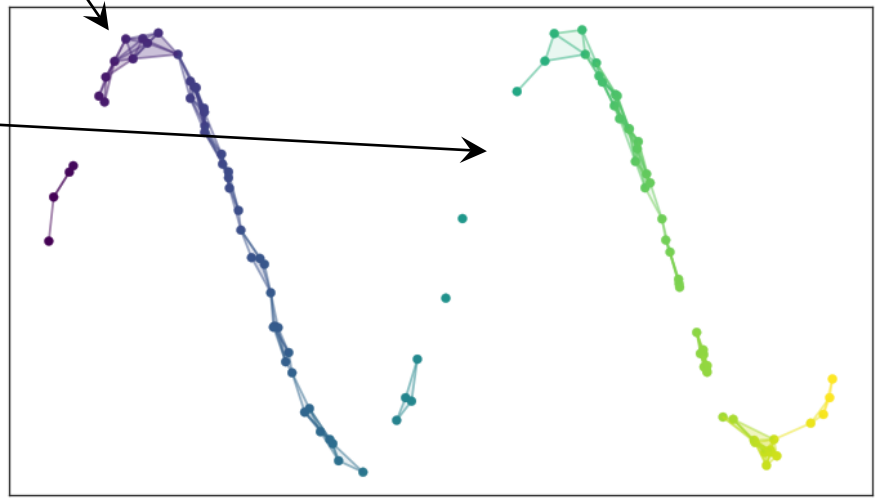


UMAP idea

- We can connect points that are of a fixed distance apart to build the objects that will allow us to recover the manifold

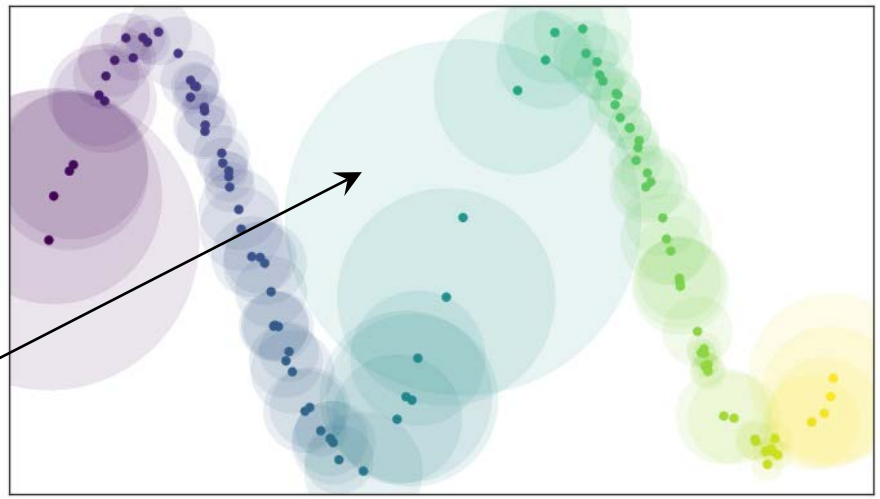


- But that will result in a lot of **missing regions** in the manifold

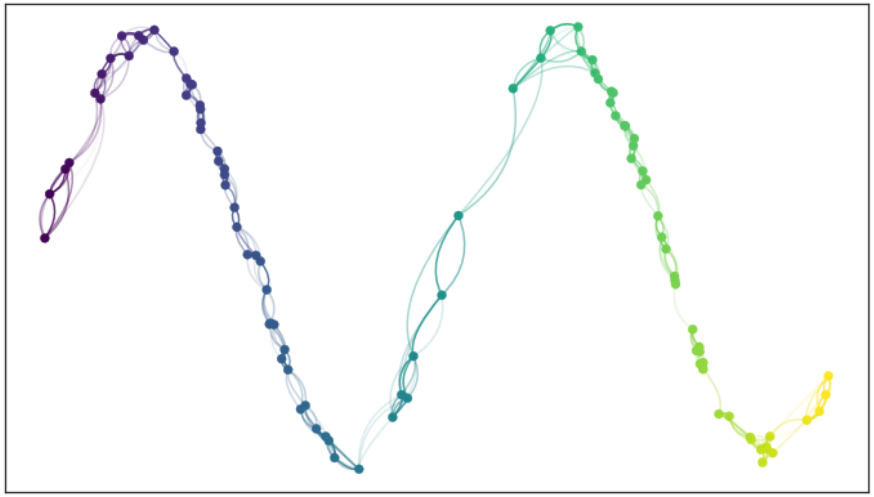


UMAP idea

- This can be solved if the distances are not treated as equal on the manifold



- The resultant objects will be able to cover the manifold



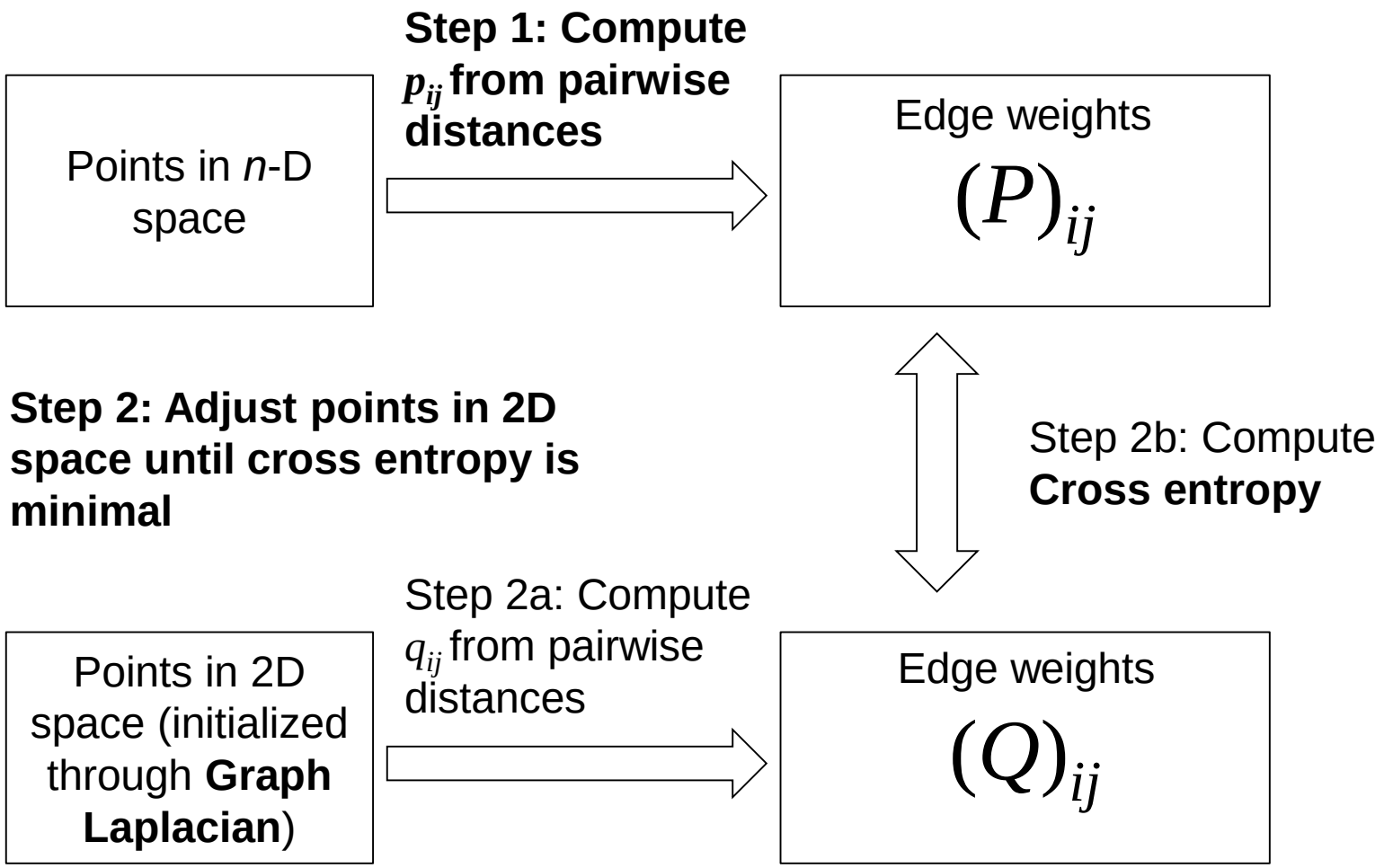
UMAP idea

- Like in t-SNE, UMAP describes the relationship between points with a value p_{ij}
- Like in t-SNE, the values of p_{ij} fall off according to some Gaussian-shaped function
  - Their fall-offs are both less steep in sparse region
- Like in t-SNE, the variance of the Gaussian-shape is different for different point i
  - Unlike t-SNE's $\mathrm{Perplexity}$, the definition of UMAP's variance can be better justified from theory (that is, to enable local connectivity)
- Unlike in t-SNE, we do not need to consider these p_{ij} values as probabilities
  - There is no longer any need for normalization

UMAP in a nutshell

- Convert the distances in *n*-D space between pairwise points i, j into an **edge weight** p_{ij}, assuming
  - Gaussian edge weight fall-off by distance
    - A Gaussian is assumed for each point, with variance determined through a **nearest-neighbor technique**
  - Result is matrix $(P)_{ij}$
- Reconstruct distances in 2D space (graph layout) such that they give rise to a matrix $(Q)_{ij}$ with
  - An **adjustable, t-distribution-like, edge weight fall-off** by the reconstructed distance
    - The distributions have same variance (making clusters evenly sized in the 2D space for better visualization)
  - Minimal **cross entropy** from $(P)_{ij}$

UMAP in a nutshell



UMAP details

- How to compute edge weight p_{ij}
  - $p_{ij} = p_{j|i} + p_{i|j} - p_{j|i} \cdot p_{i|j}$ (**union of edge weights**)

Follows from theory of manifold support

$$p_{j|i} = \exp\left(-\frac{\left\|x_i - x_j\right\|^2 - \rho_i}{\sigma_i}\right)$$

Gaussian without normalization

where

- ρ_i is the distance from x_i to its nearest neighbor
- σ_i is adjusted such that

$$\sum_j p_{ij} = \log_2 k$$

where k is a user-defined hyperparameter
(k is roughly the number of neighbors to consider as connected)

- Each point has a weight function defined by ρ_i and σ_i

UMAP details

- How to compute probability q_{ij}

$$q_{ij} = \frac{1}{1 + a\left(y_i - y_j\right)^{2b}}$$

which has a shape similar to t-distribution but adjustable by hyperparameters a and b

  - UMAP can auto-adjust a and b

UMAP lesson learned

- Like in t-SNE, $(P)_{ij}$ is a good candidate for a distance matrix
  - Natural elimination of edges between distant points
  - Distance is adjusted for local connectivity through expected number of neighbors k
- Compared to t-SNE, UMAP's $(P)_{ij}$ has better theoretical basis and can be computed faster
  - Many of the advantages of UMAP may not be due to its theoretical foundation but rather, the final form of its formulation, such as the choice of cross entropy as cost function, which were impossible in t-SNE due to its probabilistic interpretation

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