

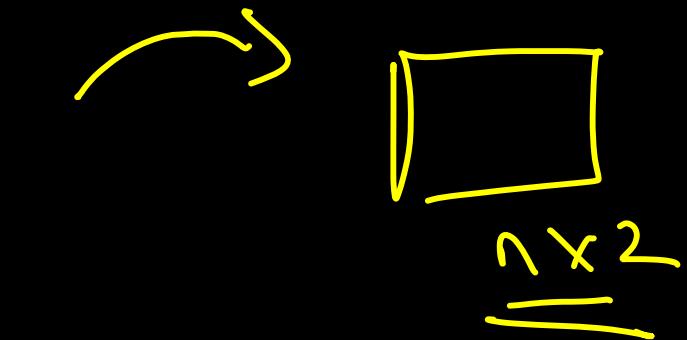
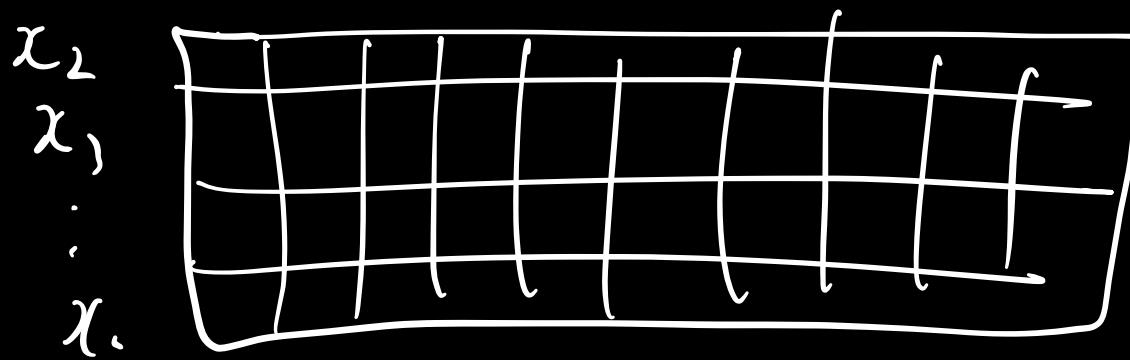
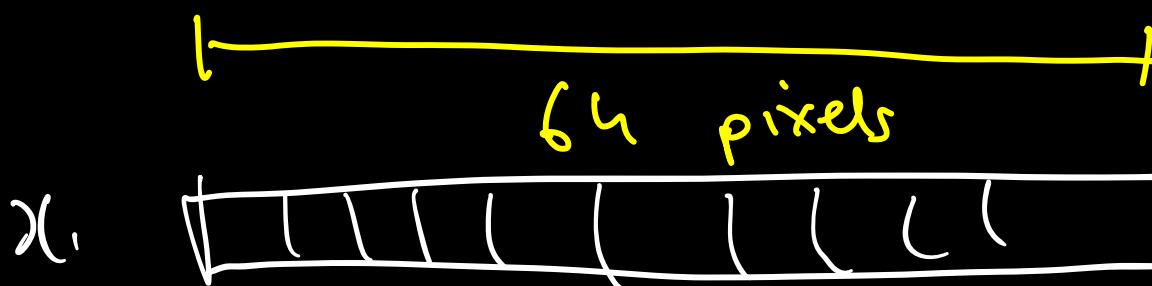
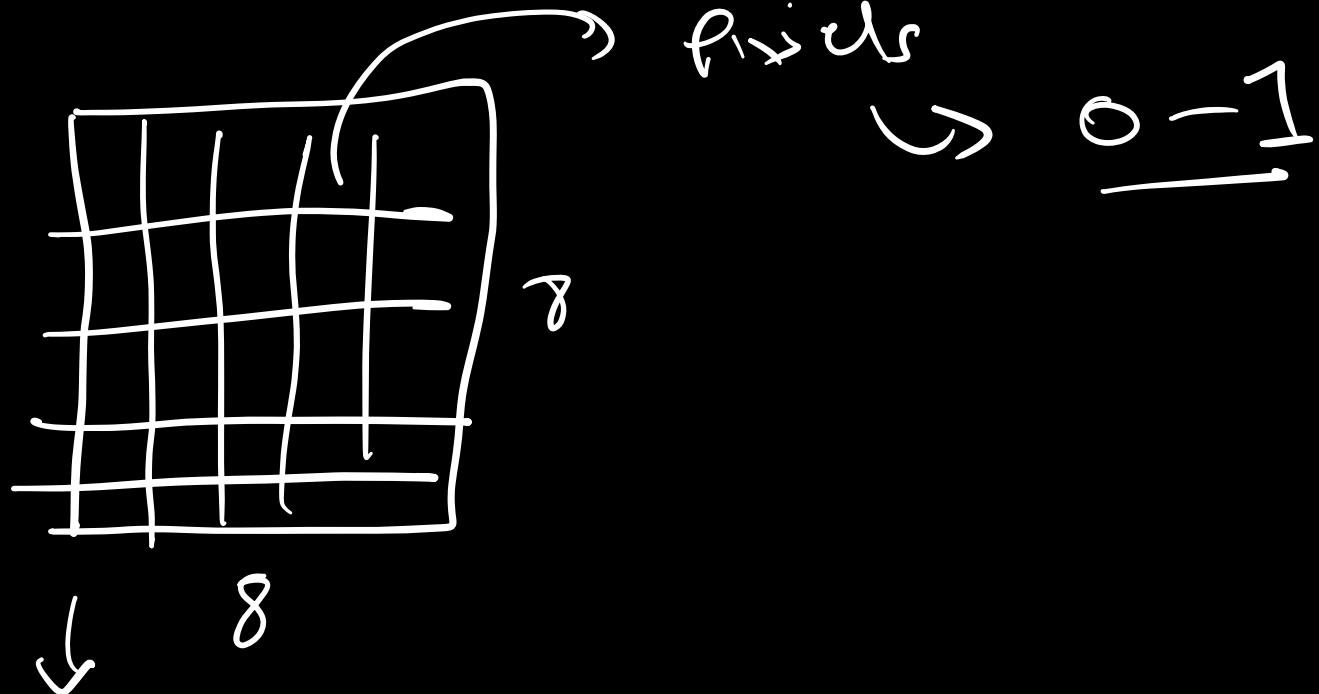
T-SNE

[Hov - 3]

→ T SNE - in depth

Recap

image \rightarrow



$n \times 64$

PCA

$$\hookrightarrow 64 \rightarrow 2$$

20% variance
preserved

$$64 \rightarrow 32$$

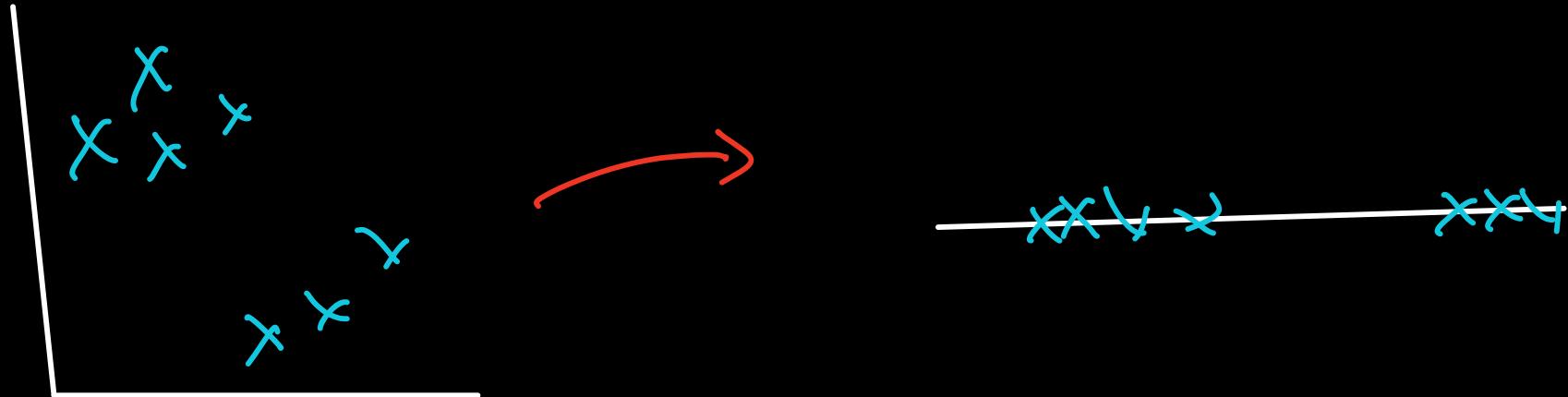
90%

$$\gamma_{64} = \frac{3\%}{=} \Rightarrow 97\% \text{ compression} \rightarrow 80\% \text{ loss}$$

$$\gamma_{32} = 50\% \rightarrow 50\% \quad || \quad \rightarrow 10\% \text{ loss}$$

Tsne

T - stochastic Neighbourhood Embedding



Preserve relative distances!

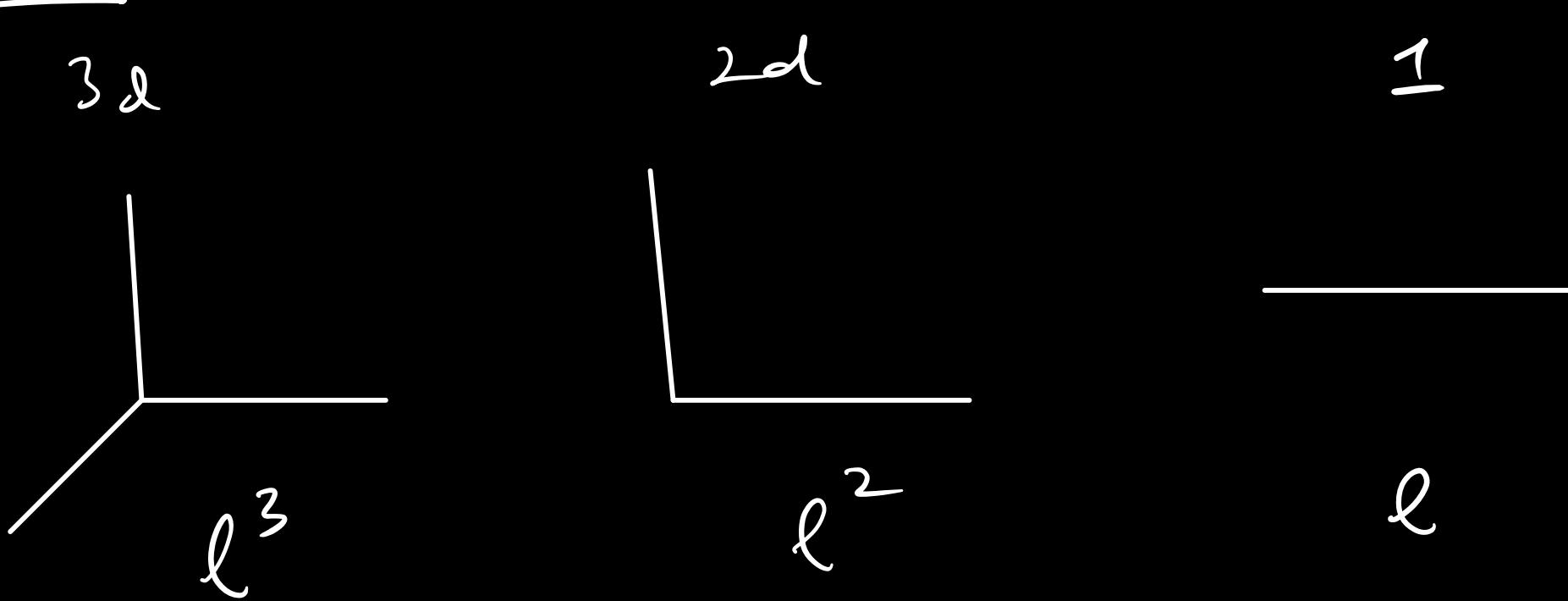
if you are my 3^{rd} closest neighbour
in d -dim, try to be my 3^{rd} closest n.
in d' -dim

Q: Can we compare distance b/w 2 points in 5d to distance in 3d?

a) Yes

b) No ✓

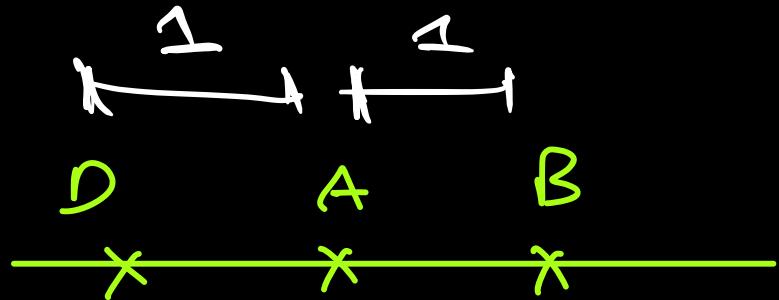
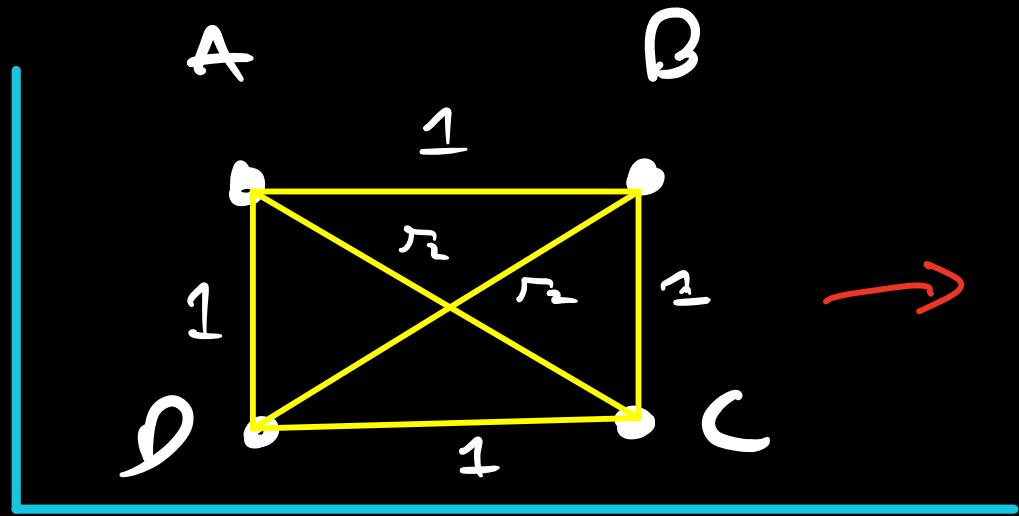
Note:



→ There is more space in high dim
↳ hence larger distances.

\Rightarrow Grounding

C?



$$d' - \dim = 1$$

$$d - \dim = 2$$

Whenever you place 'C', it will violate atleast one relationship!

So now, how do we approach this diagram?

→ Probability

↳ How can you define probability that

x_i is neighbour of x_j

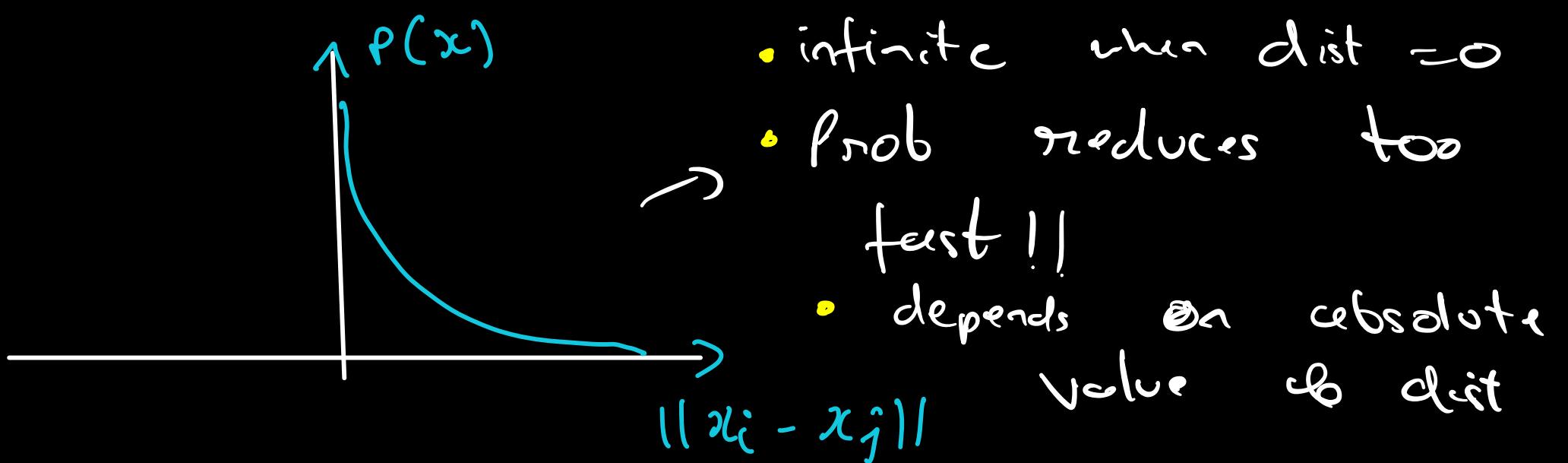
→ $\propto \sqrt{\text{distance}}$?

$$P_{ij} \propto \frac{1}{\text{dist}(x_i, x_j)} = \frac{1}{\|x_i - x_j\|}$$

(norm of $x_i - x_j$)

$$\approx \|x_i - x_j\|^{-1} \Rightarrow \text{dist}^{-1}$$

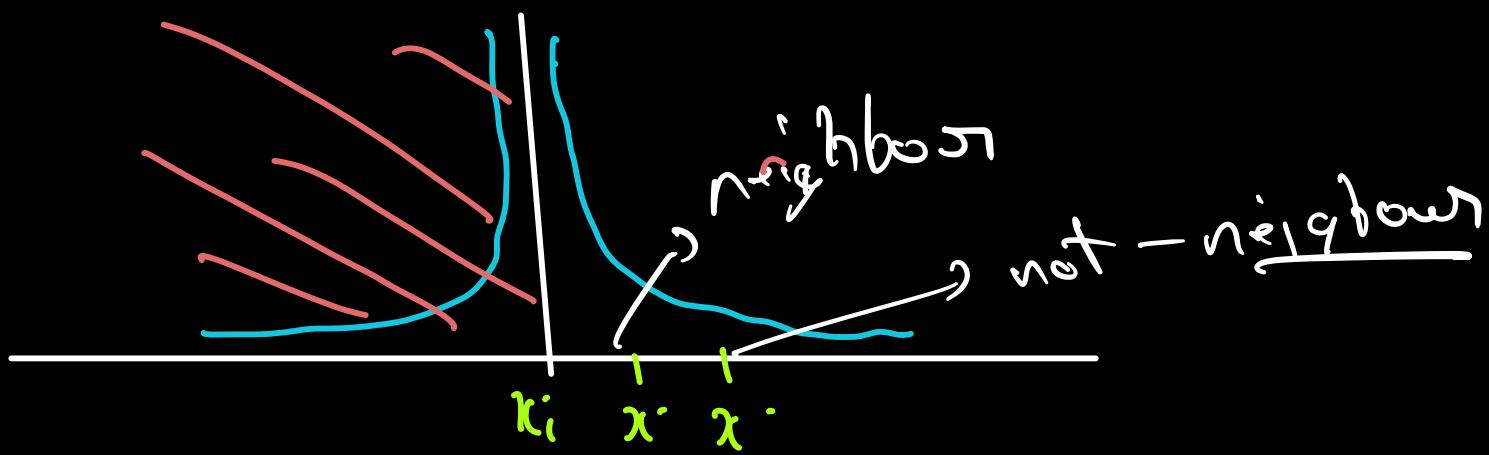
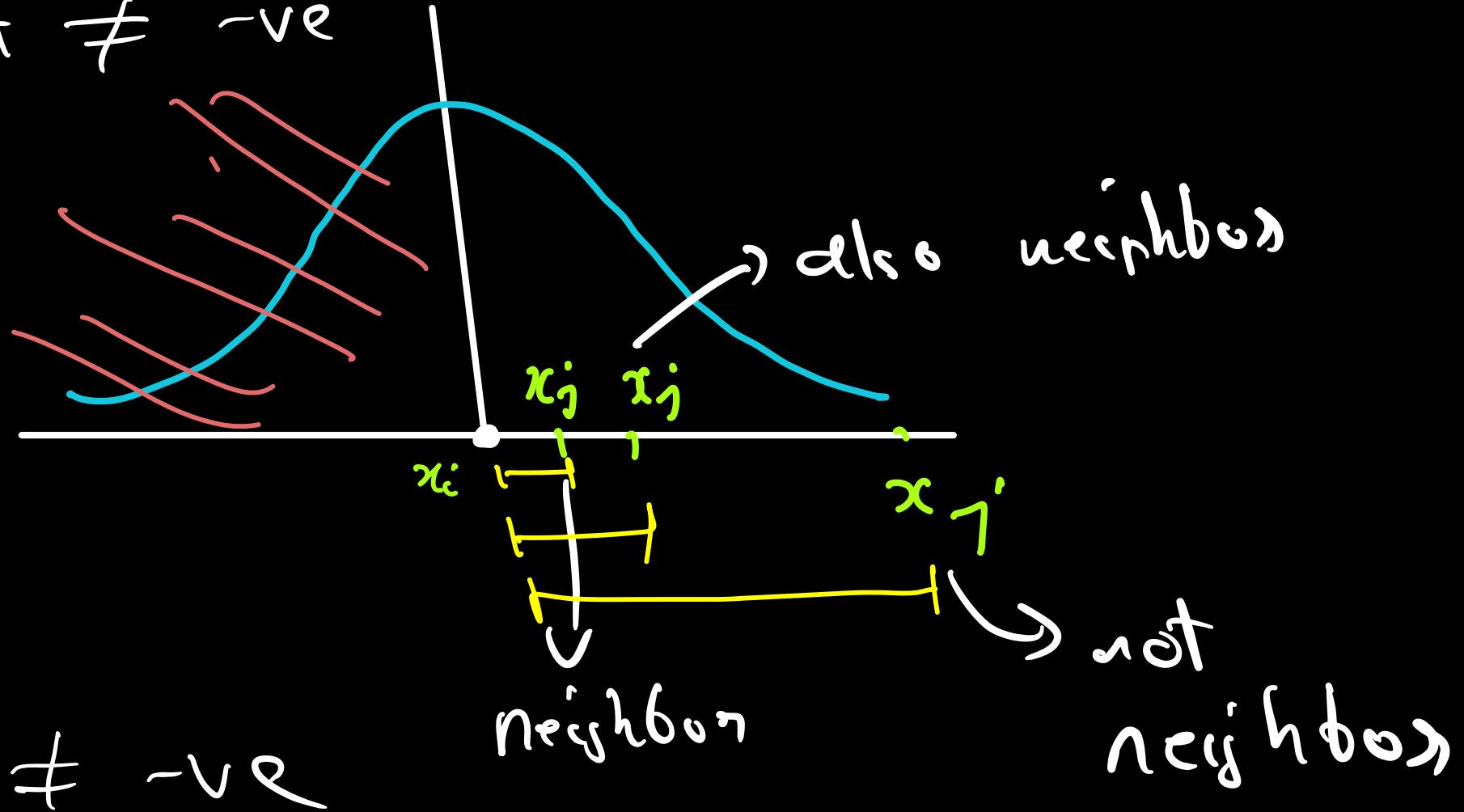
↓
how will the plot
look?

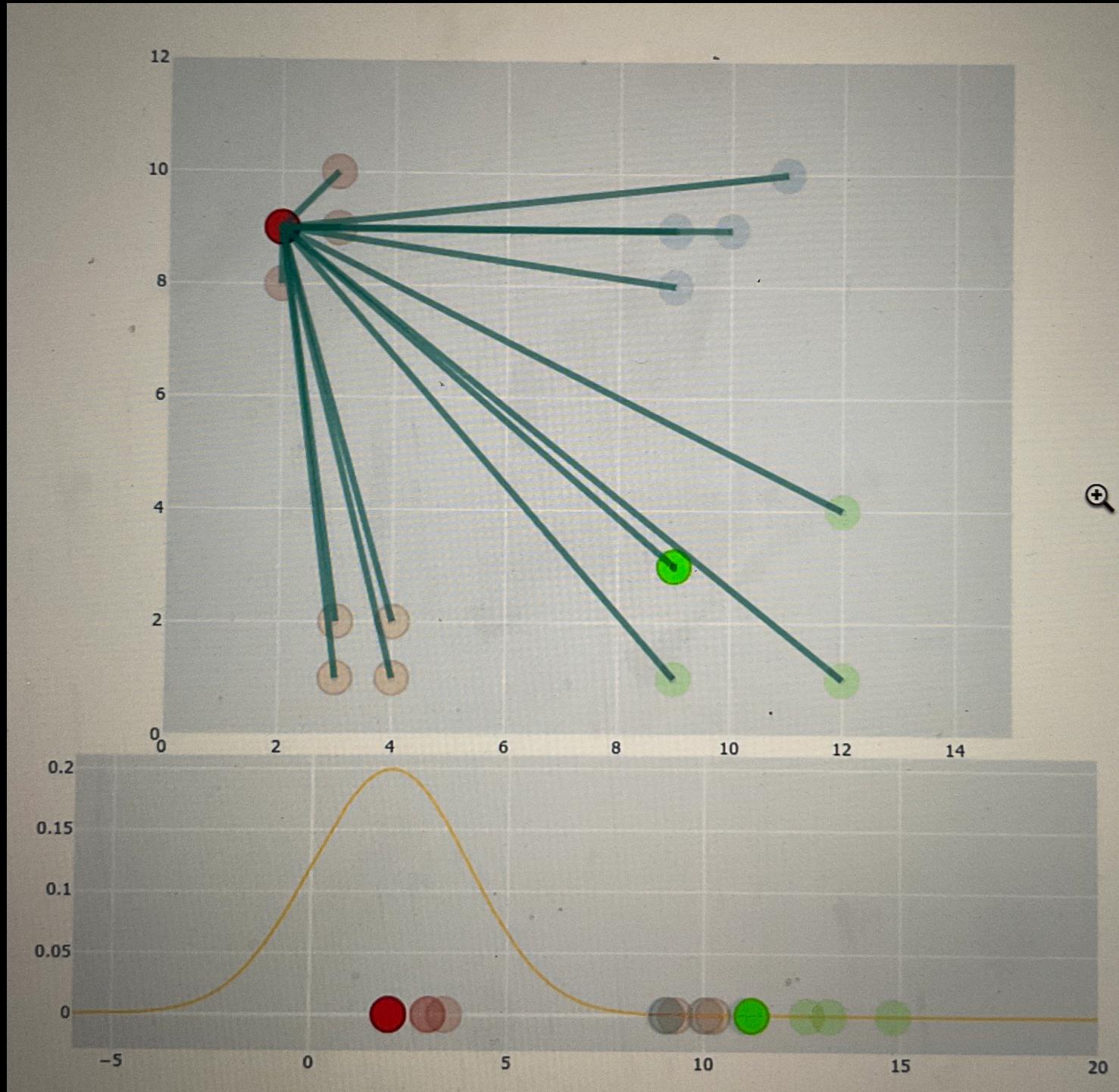


Let's take inspiration from gaussian distributions for a better pdf!

$$N(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{x-\mu}{\sigma} \right)^2}$$

$\text{dist} \neq -\text{ve}$





So let's define probability as:

$$P_{ij} = \frac{e^{-\|x_i - x_j\|^2 / 2\sigma_i^2}}{\sum_{k \neq i} e^{-\|x_i - x_k\|^2 / 2\sigma_i^2}}$$

sort of gaussian



denominator to
make

$$\sum_i \sum_j P_{ij} =$$

→ Paper

We may also write this as:

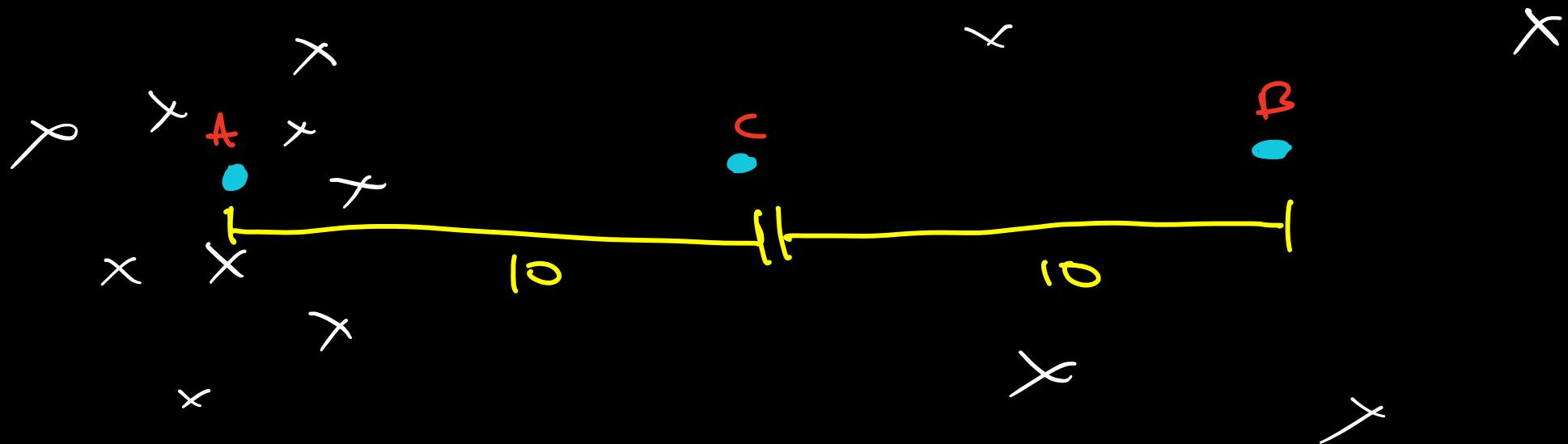
dist of x_i to x_j

$$P_{ij} = \frac{e^{-d_{ij}^2 / 2\sigma_i^2}}{\sum \text{sum of all } P_{ij}}$$

↑
estimator of
local density

- The std w.r.t each x_i creates a scaled distance effect in the neighbourhood of x_i [not absolute]
- denormalization ensures $\sum P_{ij} = 1$

Ques:



- a) $P(A, B) = P(B, C)$
- b) $P(A, B) > P(B, C)$
- c) $P(A, B) < P(B, C)$ ✓

Since 'A' has more local density, C gets lower prob of being its neighbour.

2. Stochastic Neighbor Embedding

Pg: 2581

Stochastic Neighbor Embedding (SNE) starts by converting the high-dimensional Euclidean distances between datapoints into conditional probabilities that represent similarities.¹ The similarity of datapoint x_j to datapoint x_i is the conditional probability, $p_{j|i}$, that x_i would pick x_j as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian centered at x_i . For nearby datapoints, $p_{j|i}$ is relatively high, whereas for widely separated datapoints, $p_{j|i}$ will be almost infinitesimal (for reasonable values of the variance of the Gaussian, σ_i). Mathematically, the conditional probability $p_{j|i}$ is given by

Pg: 2583

The remaining parameter to be selected is the variance σ_i of the Gaussian that is centered over each high-dimensional datapoint, x_i . It is not likely that there is a single value of σ_i that is optimal for all datapoints in the data set because the density of the data is likely to vary. In dense regions, a smaller value of σ_i is usually more appropriate than in sparser regions. Any particular value of σ_i induces a probability distribution, P_i , over all of the other datapoints. This distribution has an entropy which increases as σ_i increases. SNE performs a binary search for the value of σ_i that produces a P_i with a fixed perplexity that is specified by the user.³ The perplexity is defined as

Perplexity

↪ how many neighbouring points to use to define neighbourhood (i.e compute $\underline{\sigma}_i$)
→ V. imp will cover in detail.

produces a P_i with a fixed perplexity that is specified by the user.³ The perplexity is defined as

$$Perp(P_i) = 2^{H(P_i)}, \quad \hookrightarrow \text{Hyperparam}$$

where $H(P_i)$ is the Shannon entropy of P_i measured in bits

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

The perplexity can be interpreted as a smooth measure of the effective number of neighbors. The performance of SNE is fairly robust to changes in the perplexity, and typical values are between 5 and 50.

The minimization of the cost function in Equation 2 is performed using a gradient descent method. The gradient has a surprisingly simple form

Perplexity $\propto \underline{\sigma}_i$

Similarly, in the d' dim,

$$\Omega_{ij} = \frac{e^{-\|y_i - y_j\|^2}}{\sum} \quad (\text{Here there is no } \sigma, \text{ i.e. } 2\sigma^2 = 1)$$

where

$$x_i \xrightarrow{} y_i$$

d -dim d' -dim

Now, to get a useful d' transformation
we need

$$\rho_{ij} \approx \Omega_{ij}$$

Pg: 2581
Footnote - 2

Probability of x_i and x_j being neighbours
must be as close as possible to prob

of y_i and y_j being neighbours

Q: What do we usually do next?

→ Optimisation

Q: What do we need?

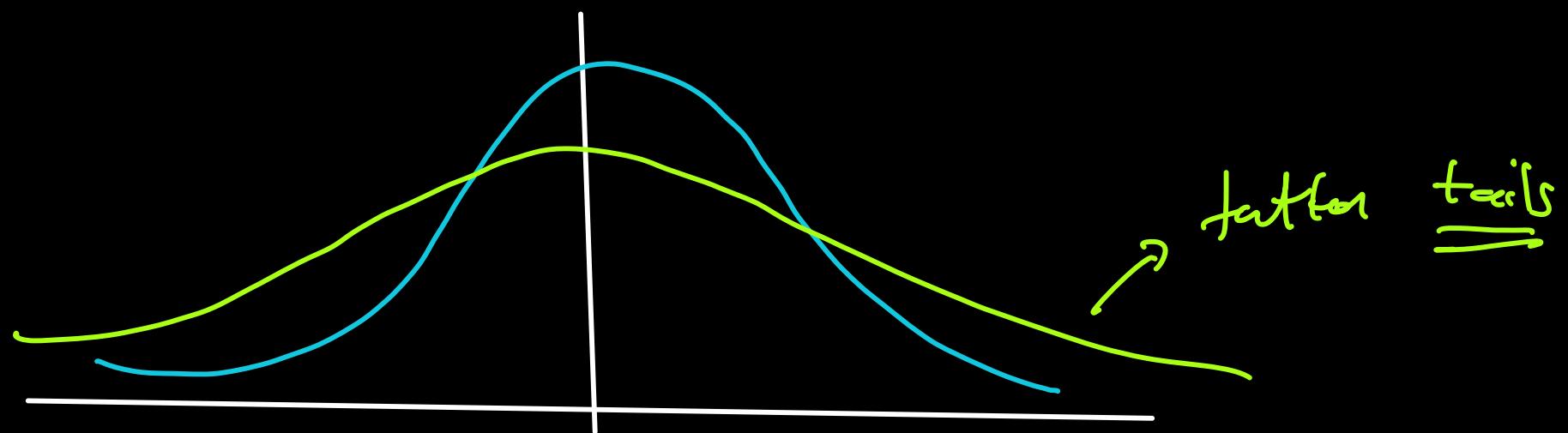
→ Loss funcⁿ

Q: Can you think of some metric
that compares distributions?

→ KS test statistic

T-distribution

↳ Researchers proposed the use of T-dist instead of 'gaussian', with $dof=1$ for d -dim



This is able to capture distant neighbours better. \rightarrow Allow a bit larger distances in d dimensions

3.3 Mismatched Tails can Compensate for Mismatched Dimensionalities

Since symmetric SNE is actually matching the joint probabilities of pairs of datapoints in the high-dimensional and the low-dimensional spaces rather than their distances, we have a natural way of alleviating the crowding problem that works as follows. In the high-dimensional space, we convert distances into probabilities using a Gaussian distribution. In the low-dimensional map, we can use a probability distribution that has much heavier tails than a Gaussian to convert distances into probabilities. This allows a moderate distance in the high-dimensional space to be faithfully modeled by a much larger distance in the map and, as a result, it eliminates the unwanted attractive forces between map points that represent moderately dissimilar datapoints.

In t-SNE, we employ a Student t-distribution with one degree of freedom (which is the same as a Cauchy distribution) as the heavy-tailed distribution in the low-dimensional map. Using this distribution, the joint probabilities q_{ij} are defined as

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}. \quad (4)$$

We use a Student t-distribution with one degree of freedom (which is the same as a Cauchy distribution) as the heavy-tailed distribution in the low-dimensional map. Using this distribution, the joint probabilities q_{ij} are defined as

tances from i to the other datapoints in the two-dimensional map, we get the following “crowding problem”: the area of the two-dimensional map that is available to accommodate moderately distant datapoints will not be nearly large enough compared with the area available to accommodate nearby datapoints. Hence, if we want to model the small distances accurately in the map, most of the points

that are at a moderate distance from datapoint i will have to be placed much too far away in the two-dimensional map.) In SNE, the spring connecting datapoint i to each of these too-distant map

Hence,

$$Q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum}$$

KL divergence

Here we use a new loss function,

$$\text{KL Div}(P, Q) = \sum_i \sum_j P_{ij} \log \left(\frac{P_{ij}}{Q_{ij}} \right)$$

≠ -ve,
proof in
post next.

if x_i and x_j
are far away
 ≈ 0 we don't

large if
they are
not the

\Rightarrow
case!

→ it's hard for p_{ij} to be smaller
for nearby points because q_{ij} comes
from T-dist and p_{ij} comes from
gaussian! Hence, this won't be -ve
easily (very small -ve values)

KL Div is used to compare distributions

Finally, to get new coordinates, we do:

$$\min_{Y} \sum_i \sum_j p_{ij} \log\left(\frac{p_{ij}}{\alpha_{ij}}\right)$$

where, $p_{ij} = \frac{\exp(-\|x_i - x_j\|^2) / 2\sigma_i^2}{\sum}$

$$\alpha_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum}$$

Optimisation

Q: Any ideas to proceed further?

$$\min_{\gamma} \sum_i \sum_j p_{ij} \log\left(\frac{p_{ij}}{\alpha_{ij}}\right)$$

→ Gradient Descent

Q: # of learnable params?