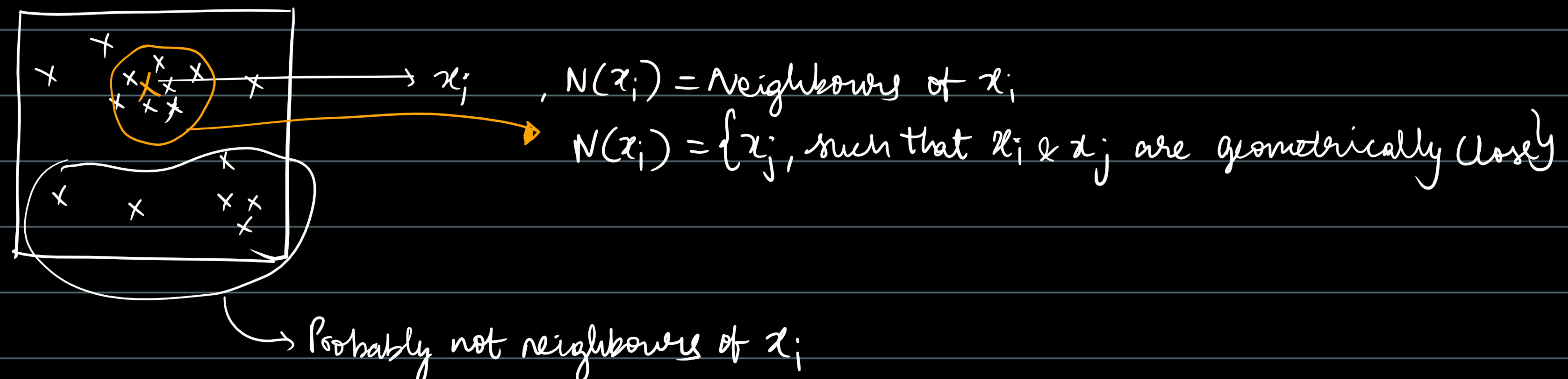


t-Dist Stochastic Neighbourhood Embedding :- (t-SNE)

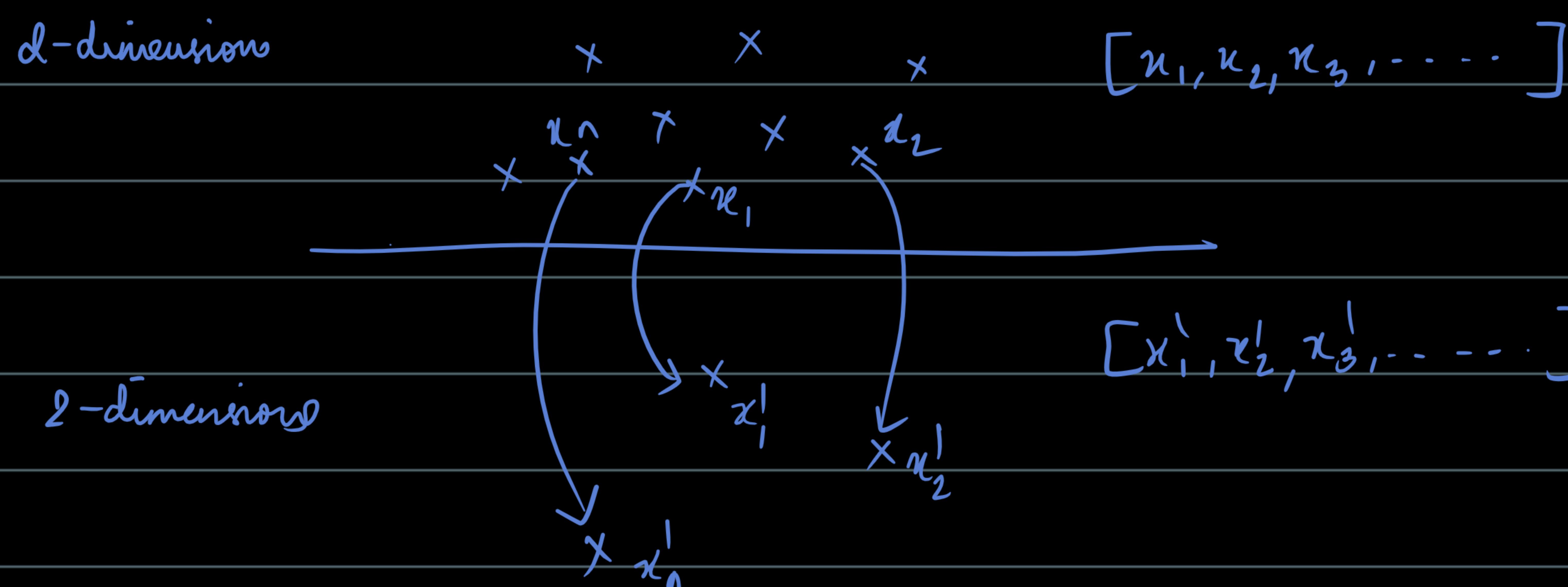
- 2008 first paper (very new)
- one of the best dimensionality reduction techniques (especially for visualization)
- Multi dimensional scaling (MDS), Sammon Mapping, Graph based techniques are some of the other techniques -
- d dimensions $\xrightarrow{t\text{-SNE}}$ 2D
- PCA tries to preserve the global structure of the data while t-SNE tries to preserve the local structure of the data.
- t-SNE is great for visualization because similar items can be plotted next to each other & reduces crowding.
- PCA gives best chance of
 - ① preserving important information
 - ② generalizable results
 - ③ Efficient dimension reduction
 - ④ Visual methods

Neighborhood of a point, Embedding :-

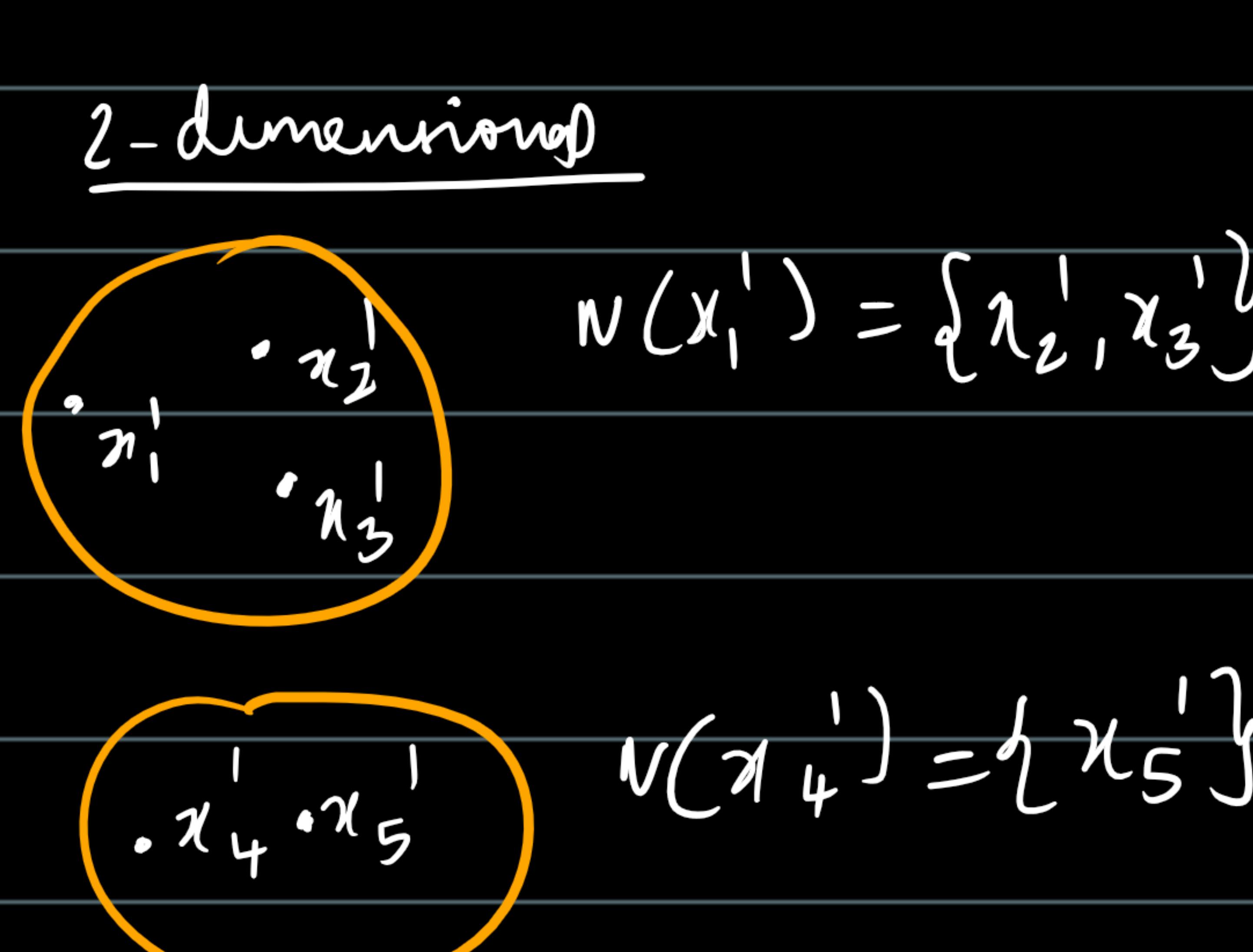
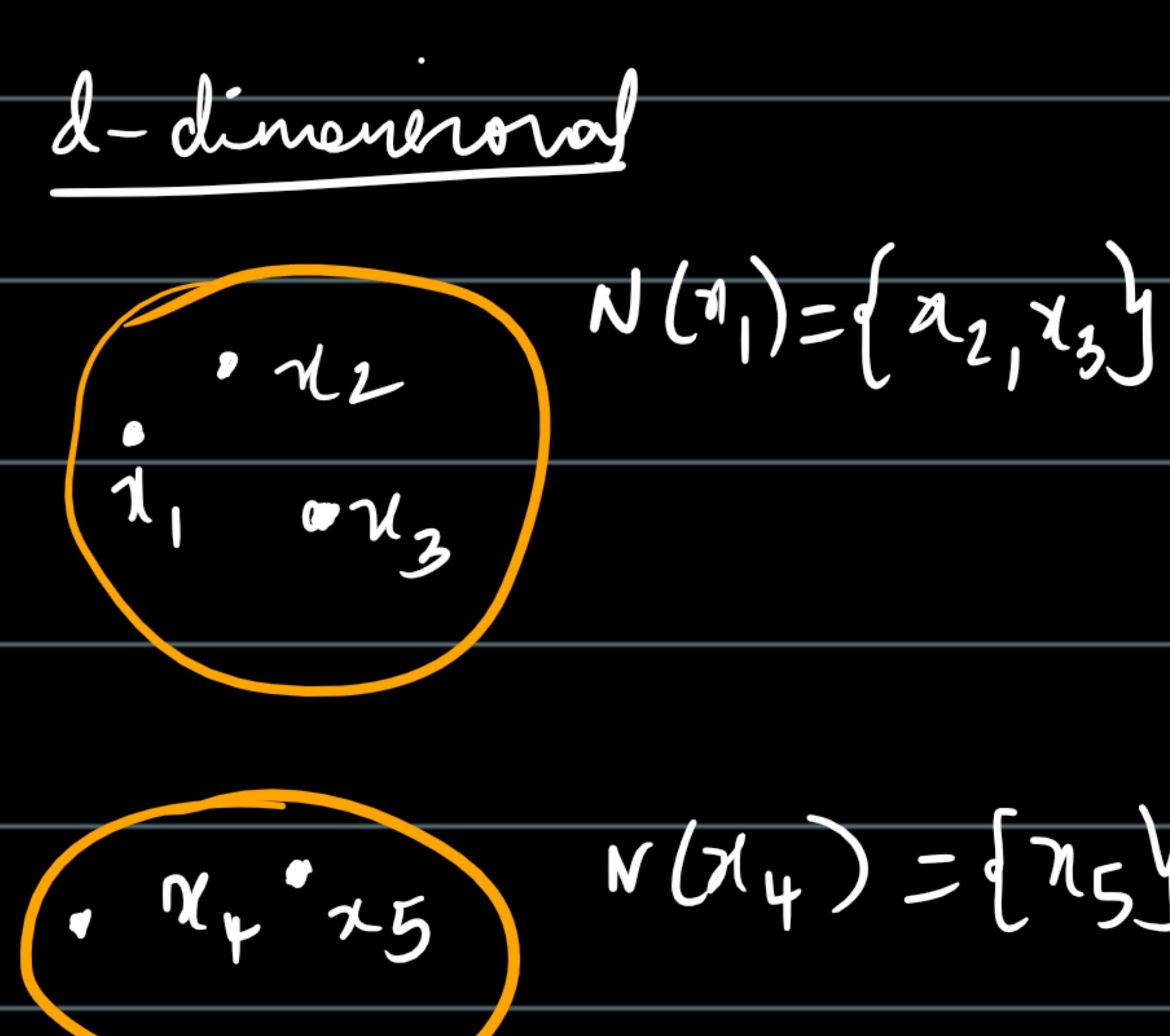
Let a d-dimensional space be



Embedding :- For every point in high dimension space we place an equivalent point in lower dimension space (2D). Process is called embedding.



Geometric Intuition of t-SNE :-



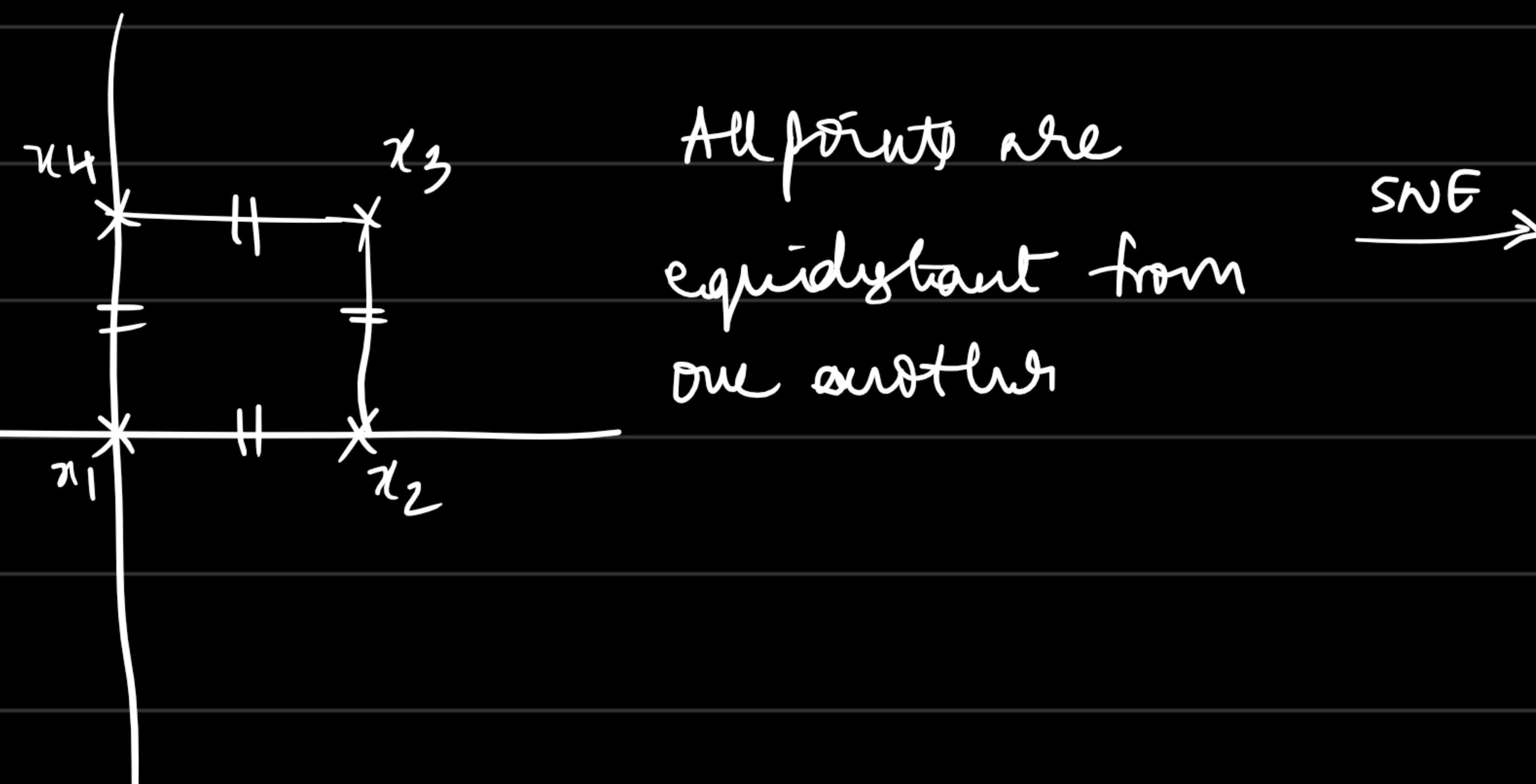
→ t-SNE tries to preserve distance in a neighbourhood.

→ $d(x_1, x_2) \approx d(x'_1, x'_2)$

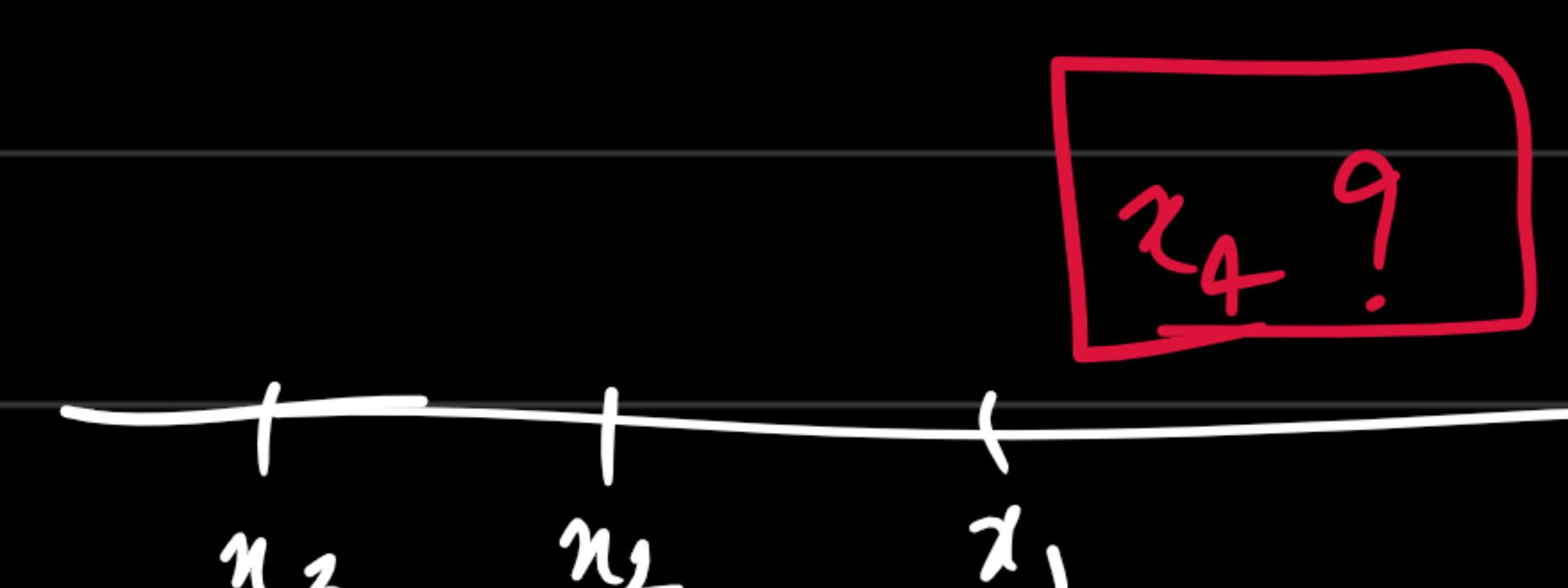
→ It does not preserve neighbours - points are going to be at the same distance as that of d-dimensions.

Crowding Problem :-

2D



1D



x4 can't be placed at an appropriate position

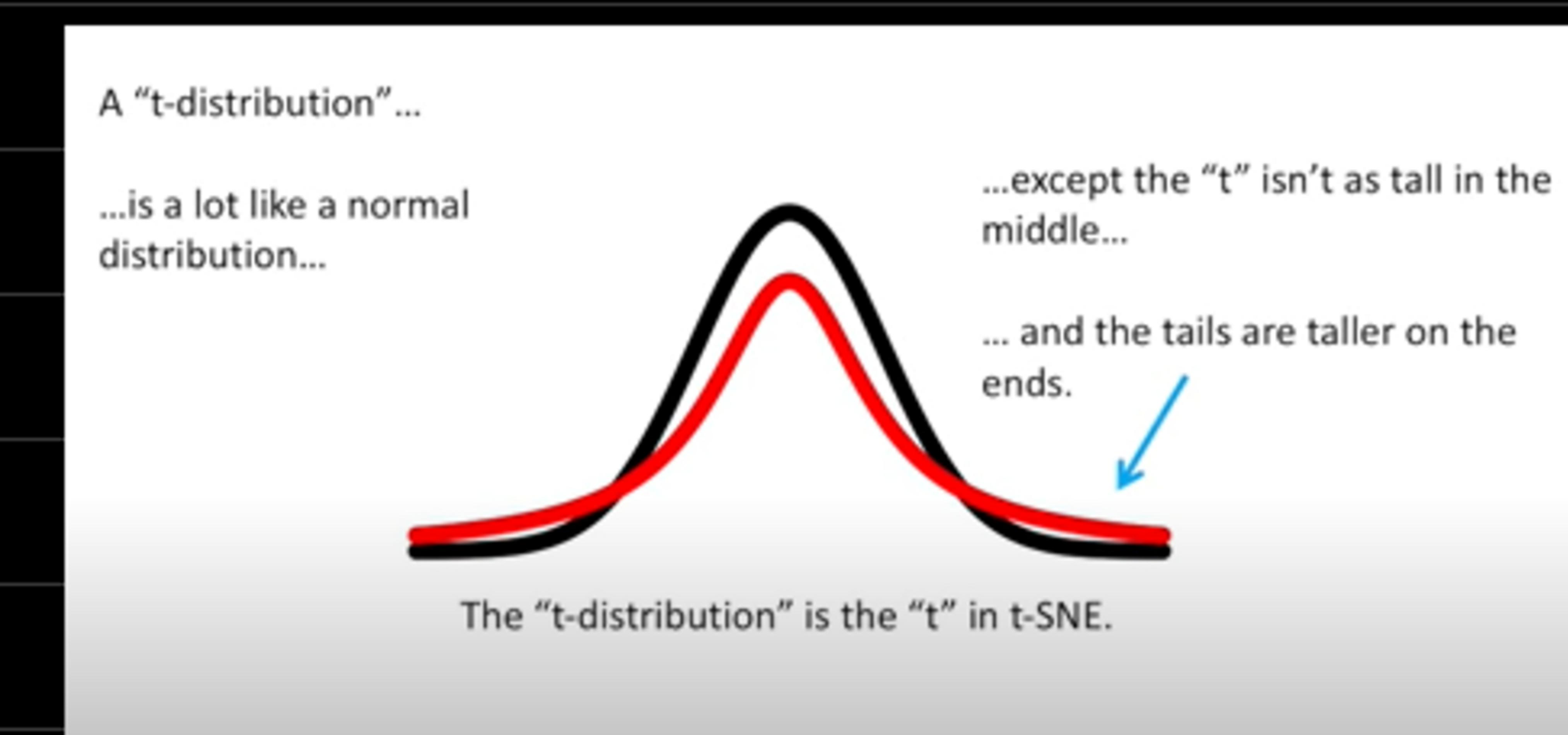
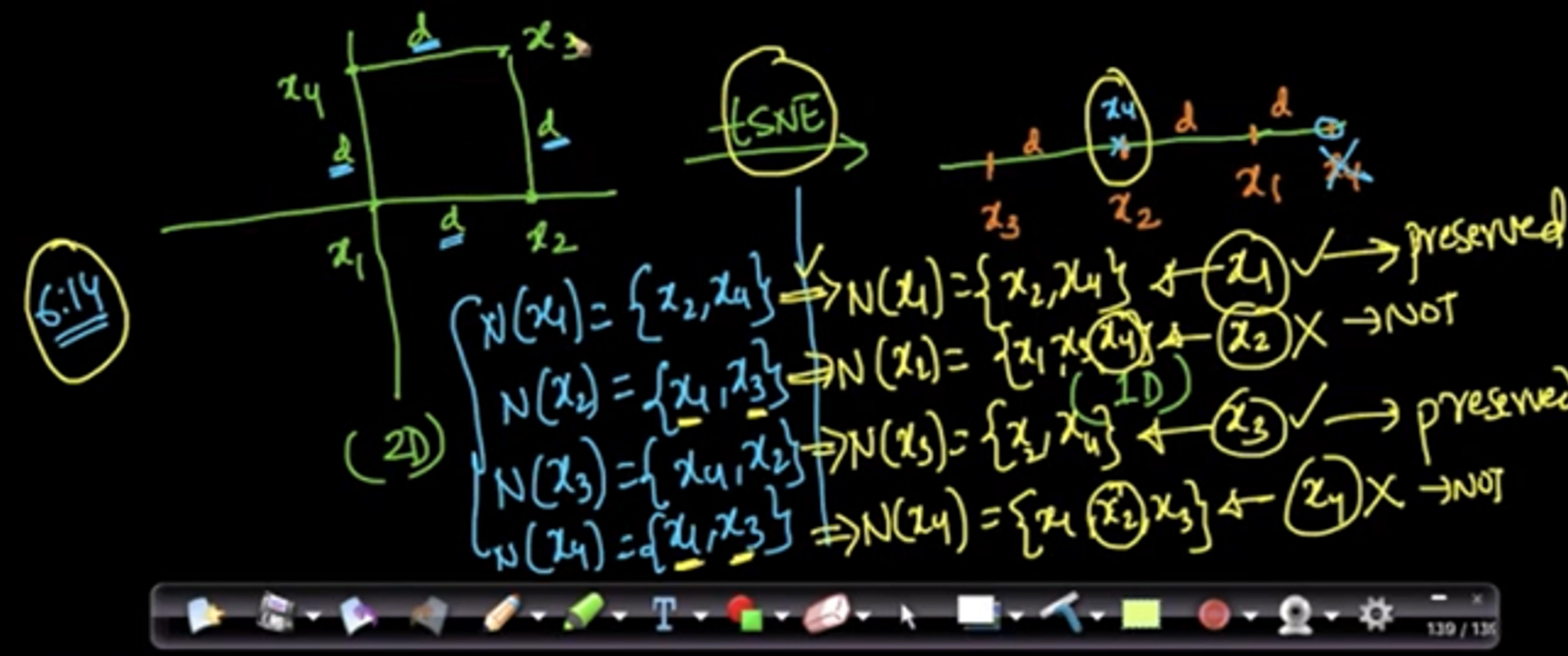
→ This is called the crowding problem.

→ SNE couldn't resolve this. t-distribution is used to resolve this problem.

Why can't points be placed on top of another point? ↴

Crowding in t-SNE

Question: (by marreddy Krishna Chaitanya Reddy @ YouTube)



→ t-distributions are used to find nearby points. t-distributions explain variance better than any other plot because it has more tail regions.

→ t-SNE is an iterative algorithm. Every iteration embedding is performed.

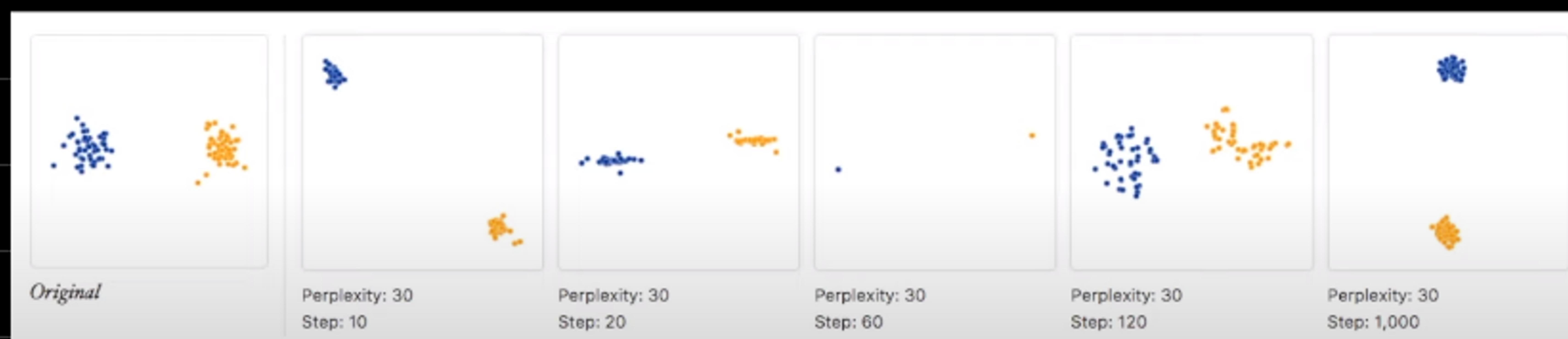
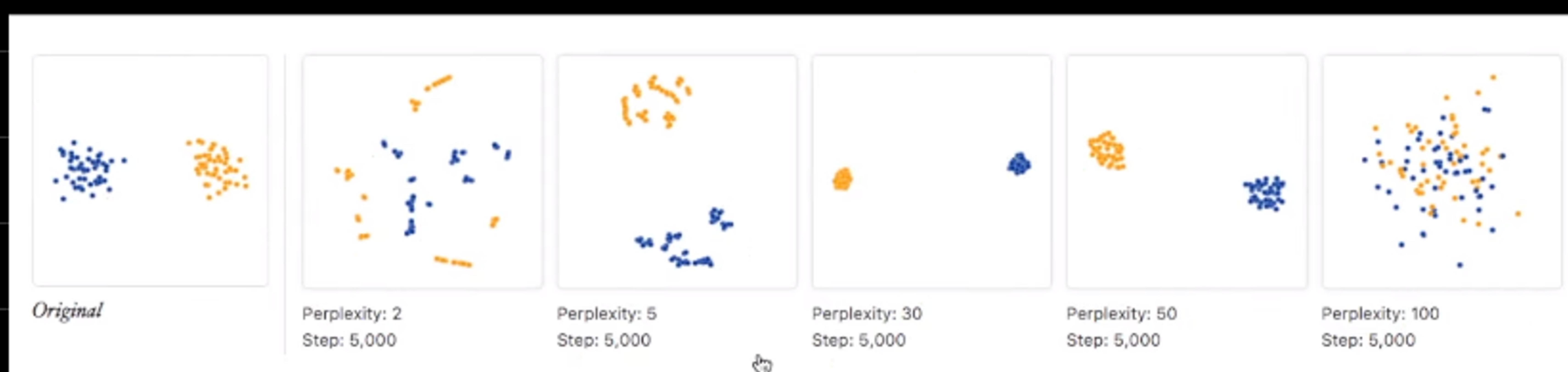
→ 2 parameters to t-SNE:-

(1) perplexity (Number of neighbors)

(2) step size (# of iteration)

→ Never run t-SNE only once. Modify/change the perplexity and the number of steps as the result varies as these values change.

→ Try to keep perplexity < # of data points.



(Keep changing step size till a figure/shape is consistent)

→ Data → Algorithm → same result every time → Deterministic Algorithm
→ different results each time → Probabilistic Algorithm.

→ t-SNE is a probabilistic Algorithm.

t-SNE

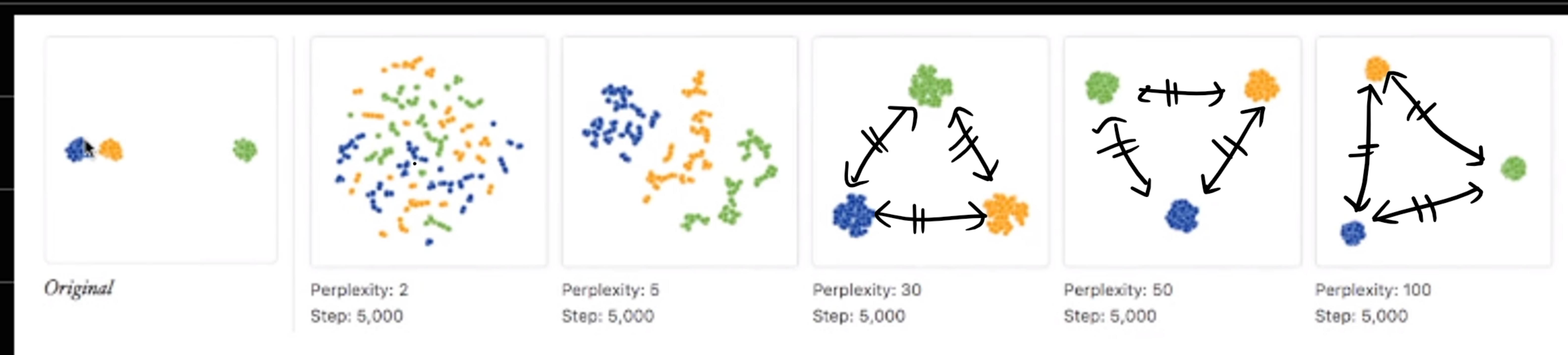
↳ Stochastic → Probabilistic

So as a result, everytime we might get diff results for same dataset & hyperparameters

→ t-SNE tries to expand dense clusters. And it tries to compress/shrink sparse clusters.

So we can't say just because two clusters are of the same size, it doesn't mean it's a good configuration

→ t-SNE does not preserve local structure.



→ Never make conclusions from random data. t-SNE might group data points into clusters when used with random data but it doesn't mean anything. Always look at diff perplexities.

→ We might see shapes in t-SNE. Never draw conclusions from one prep value. Change the values & then look at the plot.

Conclusions :-

① Run steps/iterations till shapes stabilize

② perplexity $2 \leq P < n$.

③ Never run t-SNE once. Re-run with varying perplexities & step values to see if they are stable or not. Because conclusions can't be made from just one iteration.

→ In general we run it with a large enough number of steps as the result converges at a large number of steps. Then we change perplexity and see if shape changes significantly due to the change.

→ Since t-SNE is stochastic in nature (i.e., different results each time) it's best not to use it for dimensionality reduction/classification tasks as we cannot reproduce the results.

Reason why t-SNE does not work for classification is because it does not learn from previous data.

→ Epsilon in t-SNE is the learning rate.

- t-SNE on MNIST tries to group points based on visual similarity.
- On a high level, low perplexity preserves structure on a local level & high perplexity preserve the global structure
- np.vstack(a,b) puts 'b' under 'a' like it's adding a new row to a.
- UMAP (Uniform Manifold Approximation and Projection for Dimension Reduction) is a new technique from 2018 which is faster than t-SNE but yet to be rigorously tested.
- sklearn.manifold refers to a space that is modelled in Euclidean Spaces. (t-SNE is topological space/curve in Euclidean Space).