

# Dimensionsreduktion in Merkmalsräumen



- I. Principle Component Analysis**
- 2. Multidimensional Scaling (MDS) and t-Distributed Stochastic Neighbor Embedding (t-SNE)

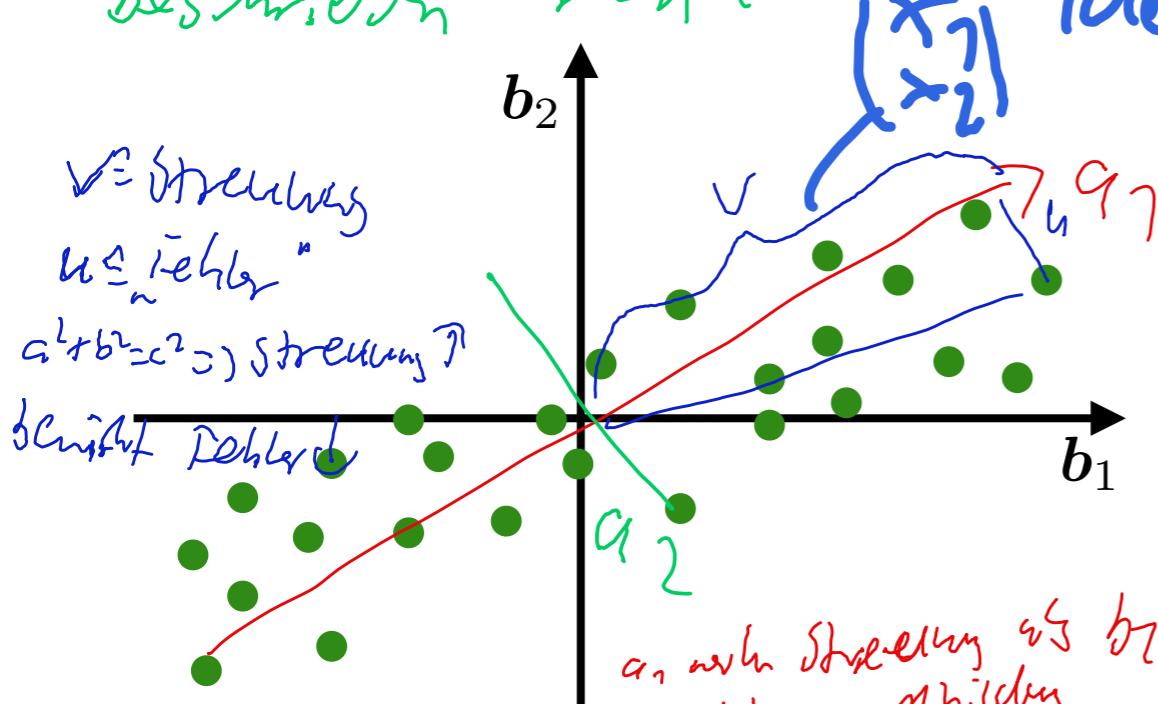
*Curse of Dimensionality*

# Principal Component Analysis

Ausgangspunkt:

hochdimensionale Merkmalsvektoren

Vektoren trotzdem gleich, können über PCA  
beschrieben werden (BWK)



$$\text{Bsp: } \mathbf{x}(b) = \mathbf{x} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} = 5 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} + 2 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_P \end{pmatrix}$$

$$\mathbf{x}(b) = \mathbf{x}_{(b)1} \cdot b_1 + \mathbf{x}_{(b)2} \cdot b_2 + \dots \mathbf{x}_{(b)P} \cdot b_P$$

orthogonal & normiert

orthogonale Vektoren  
spannen Vektorraum auf

$$-b_1 \quad b_2 \quad b_3$$

# Principal Component Analysis

Ausgangspunkt:

hochdimensionale Merkmalsvektoren

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_P \end{pmatrix}$$

$$\mathbf{x}_{(\mathbf{b})} = x_{(\mathbf{b})1} \cdot \mathbf{b}_1 + x_{(\mathbf{b})2} \cdot \mathbf{b}_2 + \dots x_{(\mathbf{b})P} \cdot \mathbf{b}_P$$

Ziel:  $\mathbf{x}_{(\mathbf{a})} = x_{(\mathbf{a})1} \cdot \mathbf{a}_1 + x_{(\mathbf{a})2} \cdot \mathbf{a}_2 + \dots x_{(\mathbf{a})P} \cdot \mathbf{a}_P$

*stehen orthogonal aufeinander*

mit  $\mathbf{a}_i^T \mathbf{a}_i = 1$  und  $\mathbf{a}_i^T \mathbf{a}_j = 0$  für  $i \neq j$

*normiert*

$$x_{(\mathbf{a})i} = \mathbf{a}_i^T \mathbf{x}_{(\mathbf{b})}$$

# Principal Component Analysis

Koordinatentransformation:

mit  $A = a_1 \ a_2 \dots a_p$ )

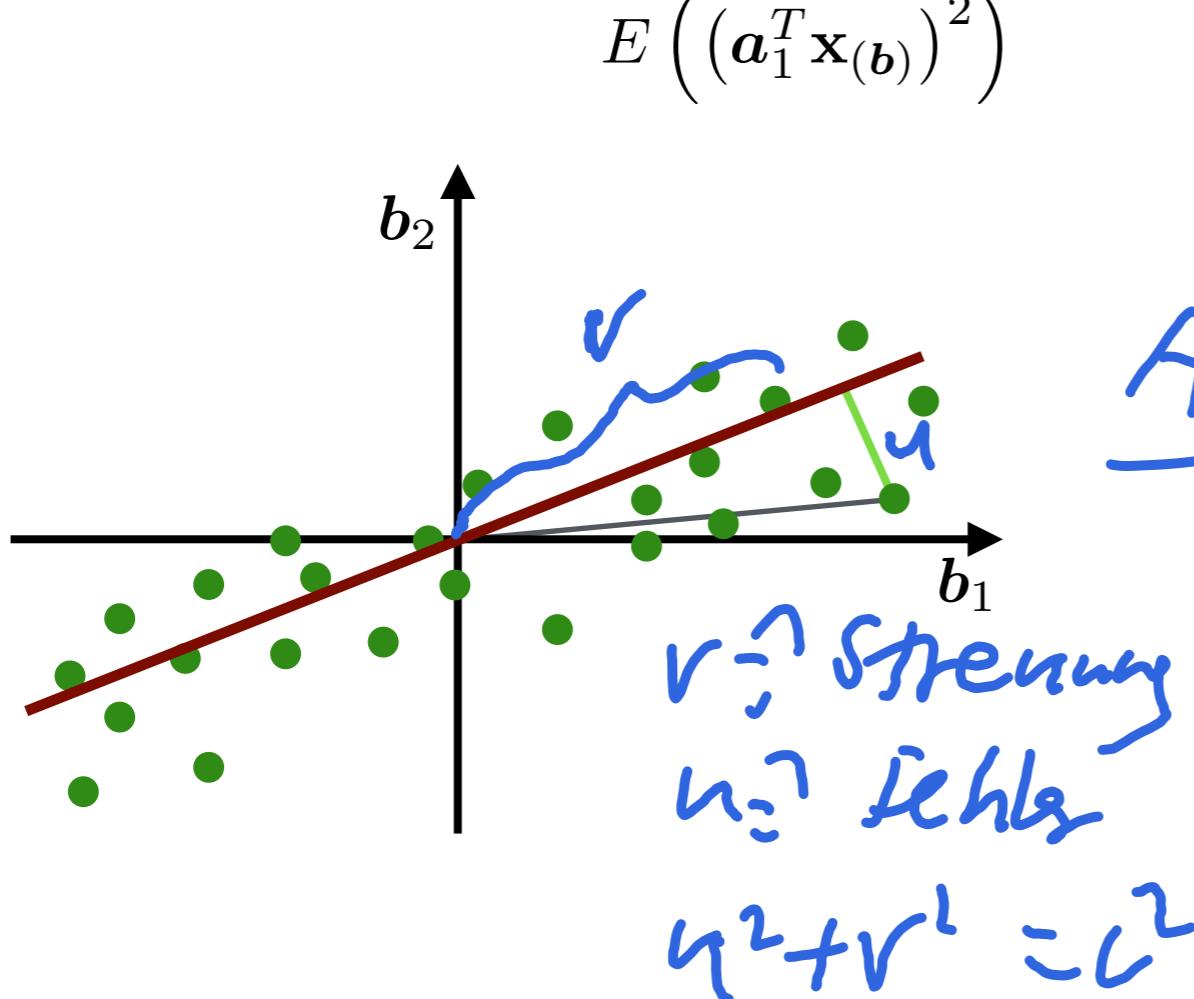
$$\mathbf{x}_{(a)} = A^T \mathbf{x}_{(b)}$$
$$\left[ \begin{array}{cccc} a_{11} & a_{21} & \dots & a_{p1} \\ a_{12} & \ddots & \ddots & \vdots \\ \vdots & & & \vdots \\ a_{1p} & \dots & \dots & a_{pp} \end{array} \right]$$

$$\mathbf{x}_{(a)i} = a_i^T \mathbf{x}_{(b)}$$

# Principal Component Analysis

Erste Hauptkomponente mit maximaler Streuung:

$$x_{(a)1} = a_1^T x_{(b)}$$



Streuung :

$$\mathbb{E}[(x - \mathbb{E}(x))^2]$$

$\approx n, \text{Teilung}$

Anh:

$$\mathbb{E}(x) = 0$$

$$\Rightarrow \mathbb{E}[(x - \mathbb{E}(x))^2]$$

$$= \mathbb{E}(x^2) - \underbrace{\mathbb{E}(x)^2}_{=0}$$

bleibt übrig

$$= \mathbb{E}((a_1^T x_{(b)})^2)$$

# Principal Component Analysis

Erste Hauptkomponente mit maximaler Streuung:

$$\mathbf{x}_{(\mathbf{a})1} = \mathbf{a}_1^T \mathbf{x}_{(\mathbf{b})}$$

$$E \left( (\mathbf{a}_1^T \mathbf{x}_{(\mathbf{b})})^2 \right)$$

$$= E \left( \mathbf{a}_1^T \mathbf{x}_{(\mathbf{b})} \mathbf{a}_1^T \mathbf{x}_{(\mathbf{b})} \right)$$

$$= E \left( \mathbf{a}_1^T \mathbf{x}_{(\mathbf{b})} \mathbf{x}_{(\mathbf{b})}^T \mathbf{a}_1 \right)$$

$$= \mathbf{a}_1^T E(\mathbf{x}_{(\mathbf{b})} \mathbf{x}_{(\mathbf{b})}^T) \mathbf{a}_1$$

$$= \mathbf{a}_1^T C \mathbf{a}_1$$

↳ Kovarianzmatrix

$$C = \begin{pmatrix} E(x_1^2) & E(x_1 \cdot x_2) & \dots \\ \vdots & E(x_2^2) & \dots \\ \ddots & \ddots & \ddots & E(x_p^2) \end{pmatrix}$$

# Principal Component Analysis

Erste Hauptkomponente mit maximaler Streuung:

$$\mathbf{x}_{(\mathbf{a})1} = \mathbf{a}_1^T \mathbf{x}_{(\mathbf{b})}$$

$$\Phi = \underbrace{\mathbf{a}_1^T C \mathbf{a}_1}_{\text{Streuung}} - \lambda (\underbrace{\mathbf{a}_1^T \mathbf{a}_1 - 1}_{\text{Länge v. Vektor mit } \mathbf{1} \text{ vergleiche}})$$

Maximieren?

$$\exists f'(\mathbf{x}) \geq 0$$

$$\frac{\partial \Phi}{\partial \mathbf{a}_1} = 2C\mathbf{a}_1 - 2\lambda\mathbf{a}_1 = 0$$

$$(\mathbf{C} - \lambda \mathbf{1})\mathbf{a}_1 = 0$$

) Ausklammern von  $\mathbf{a}_1$

Identitätsvektor

↳ Eigenwertproblem

# Principal Component Analysis

weitere Hauptkomponenten:

$$\mathbf{x}_{(a)i} = \mathbf{a}_i^T \mathbf{x}_{(b)}$$

L> nächster Schritt: finde Vektor  $a_2$  als den Vektor mit maximaler Streckung aller zu  $a_1$  orthogonale Vektoren

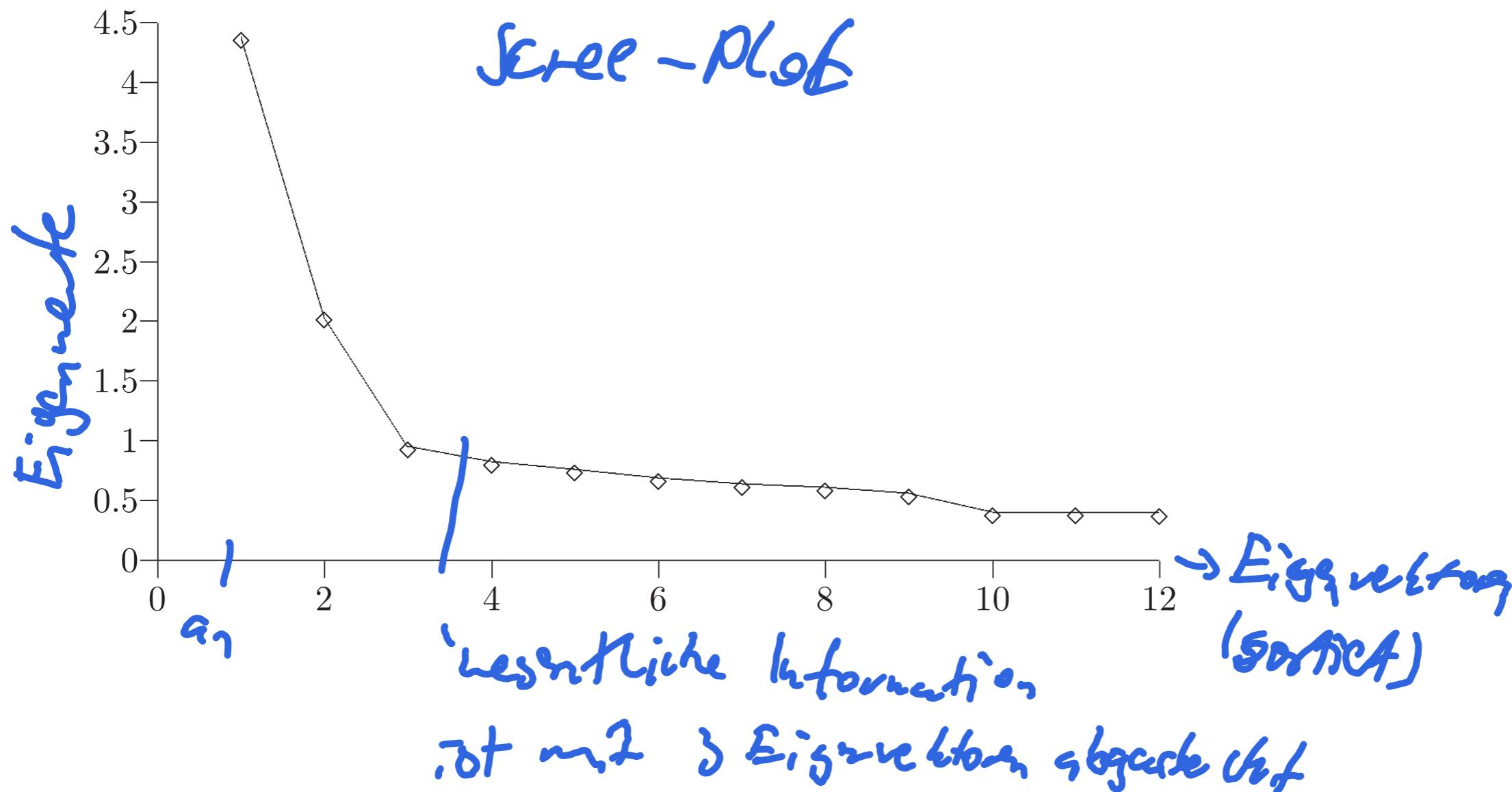
4)  $a_3 \dots a_p$  analog

$\Rightarrow$  Hauptkomponenten bilden die Eigenvektoren des DSK

5) Eigenwert  $\lambda_i$  gibt Streckung zu jenigen Basisvektor  $a_i$

# Principal Component Analysis

Reduktion der Dimension des Merkmalsraums



© Webb, Copsey

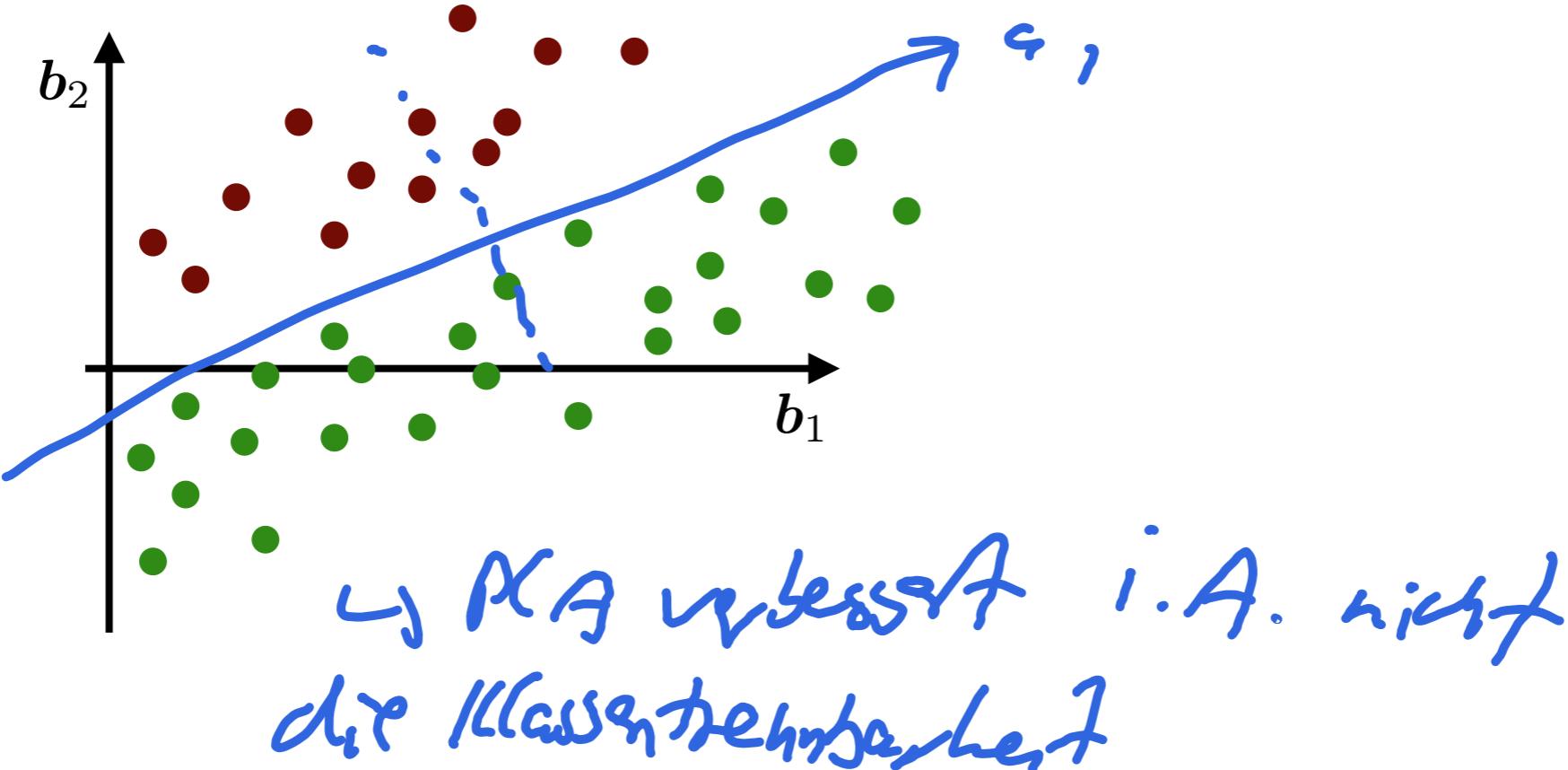
# Principal Component Analysis

Normierung

↳ Mittelwert 0, Varianz 1  
separat für jede Dimension

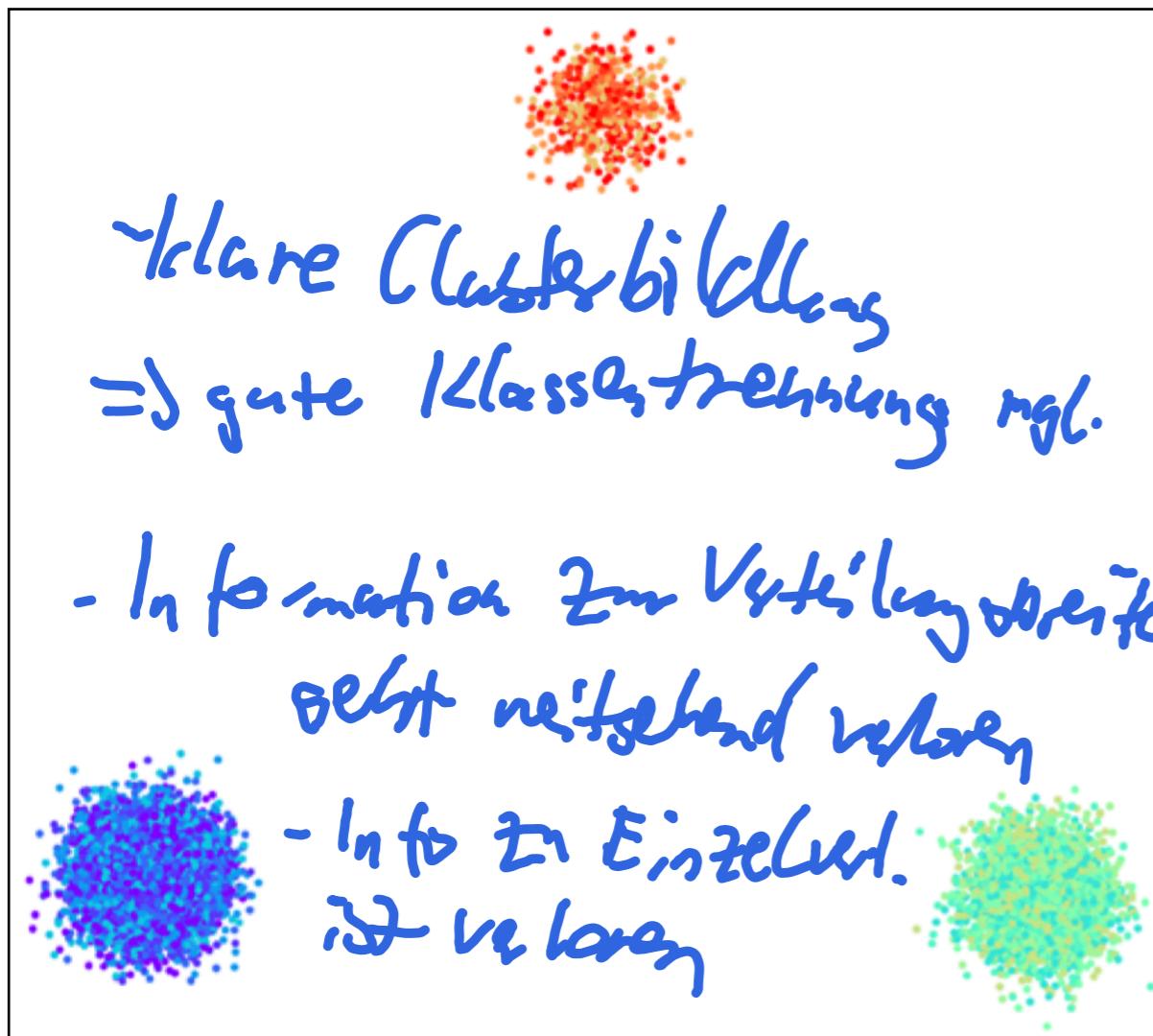
# Principal Component Analysis

Diskriminationsfähigkeit?



# Principle Component Analysis

Bsp: künstliche Daten



- 75.000 Samples gezeigt,
- PCA auf 2 Dimensionen

50-dimensionalen Geobasis,  
davon 75 Stück

3 verschiedene, nicht - überlappende  
Klassen

# Samples : rot < Grüne dann  
Klassen: Mittelwerte unterschiedl.,  
aber ähnlich innerhalb einer Klasse  
Bereik der Geobasis untersch. ist  
zwischen den Klassen

# Principle Component Analysis

Bsp: MNIST



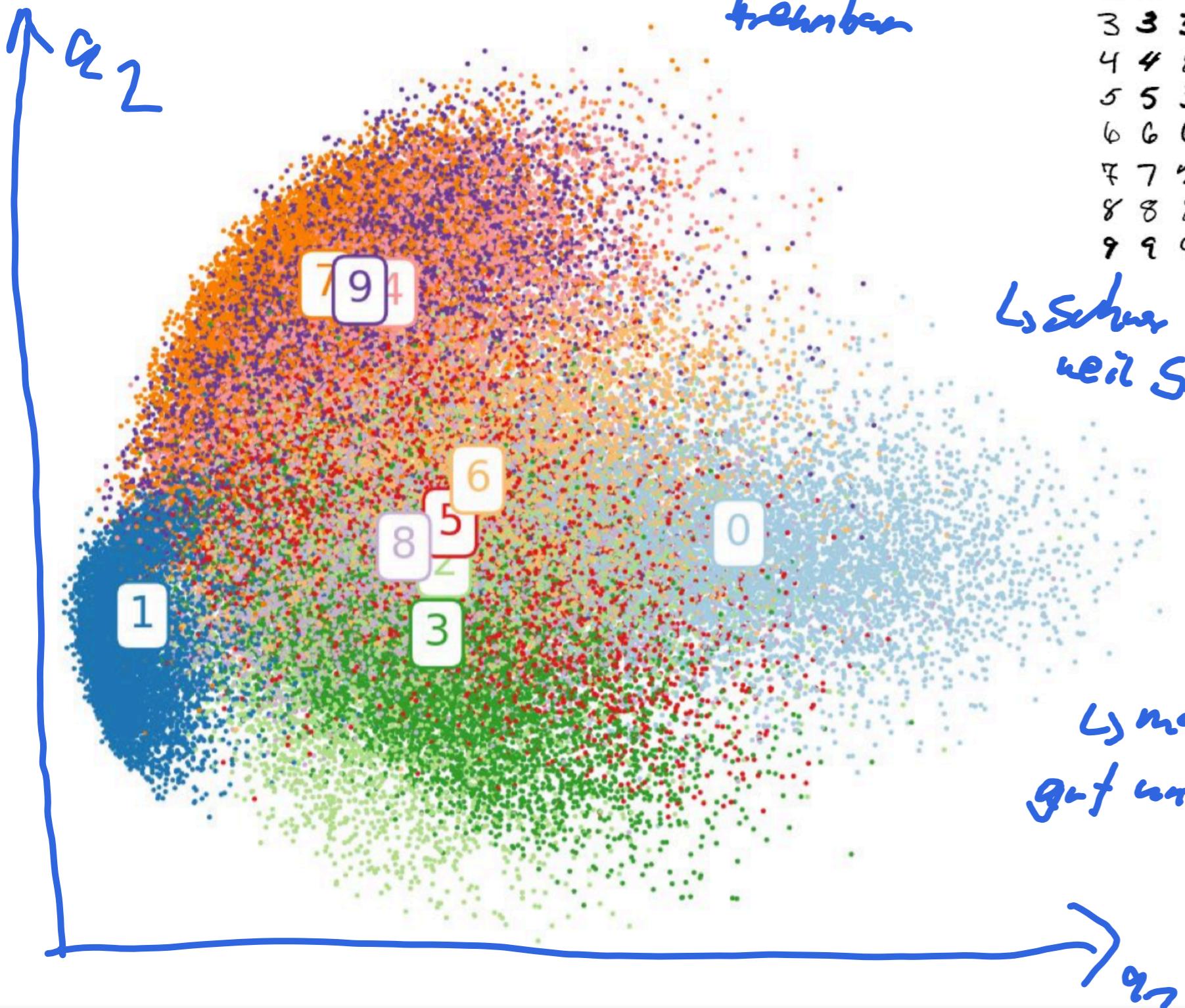
$$28 \times 28 = 784$$

jedes Pixel stellt eine Dimension dar

70.000 Samples

# Principle Component Analysis

# MNIST



Löscher zu klassifizieren,  
weil Grenzen wanken (7,9,4)

↳ manche Cluster lassen sich  
gut unterscheiden

# Dimensionsreduktion in Merkmalsräumen



1. Principle Component Analysis
2. **Multidimensional Scaling (MDS) and t-Distributed Stochastic Neighbor Embedding (t-SNE)**

# Multidimensional Scaling

## Hauptstädte Europas (Auswahl)



Flugentfernung  
LS Anzahl von 22  
Städte

Fragestellung:  
Städte so ordnen, in 2D  
dass die Abstände der  
Flugentfernung in 3D  
entsprechen

# Multidimensional Scaling

## Hauptstädte Europas (Auswahl)

	Stockholm	Lissabon	Madrid	Paris	London	Dublin	Brüssel
Stockholm		2991	2595	1545	1433	1629	1126
Lissabon	2991		504	1454	1587	1642	1714
Madrid	2595	504		1054	1265	1452	1318
Paris	1545	1454	1054		479	782	305
London	1433	1587	1265	479		288	235
Dublin	1629	1642	1452	782	288		776
Brüssel	1281	1714	1318	305	235	776	

Mit .Symmetrisch, in halbdimensionalem Raum

# Multidimensional Scaling

## Ziel

Distanzen in hoher Dimension sollen Distanzen in niedriger Dimension entsprechen

Distanzen in hochdim. Raum

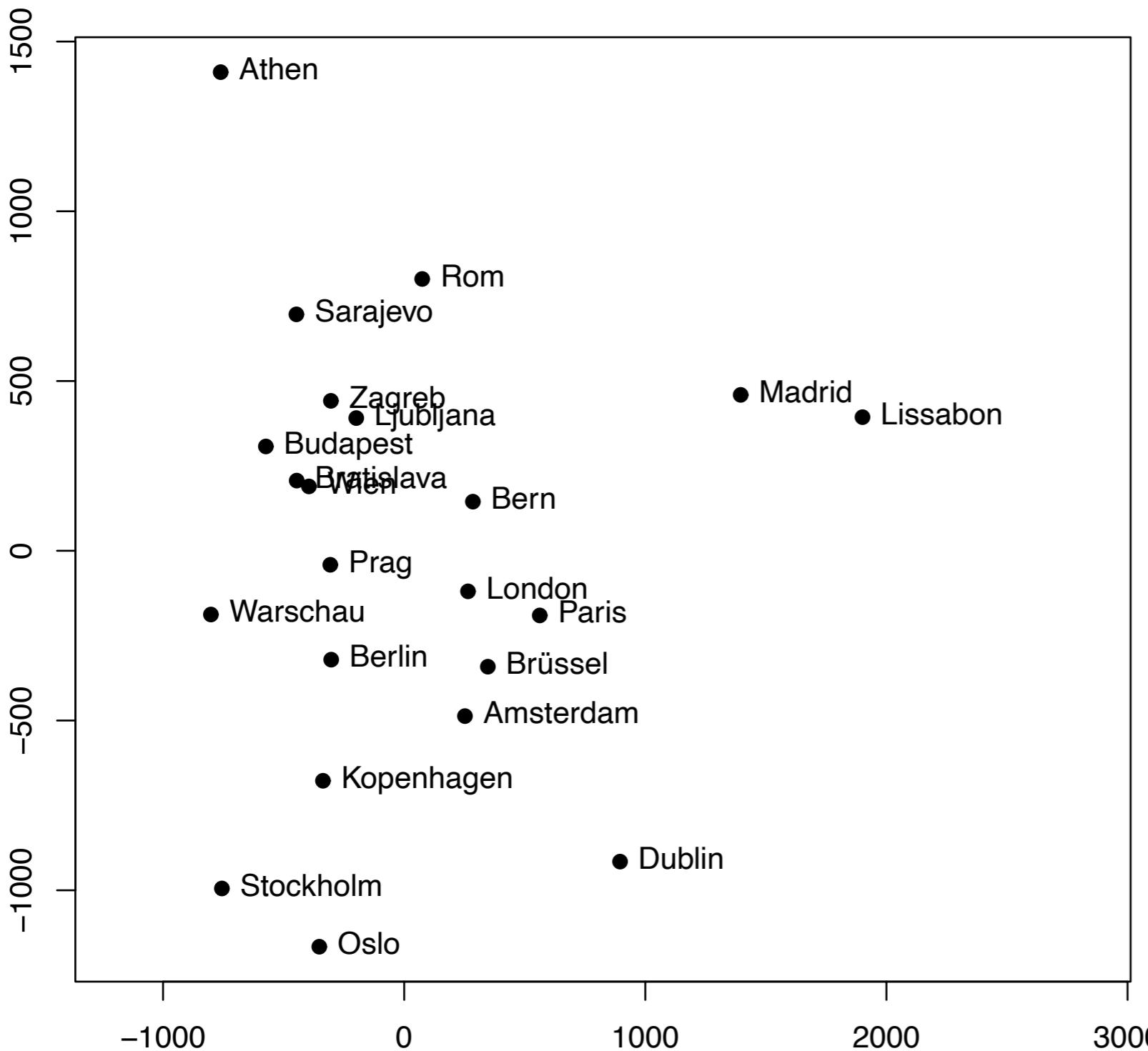
$$D_{\text{MDS}} = \sum_{i < j} (d_{ij} - \|z_i - z_j\|)^2$$

entprechende Samples im niedrig-dimensionalen Raum

> iteratives Optimierungsverfahren

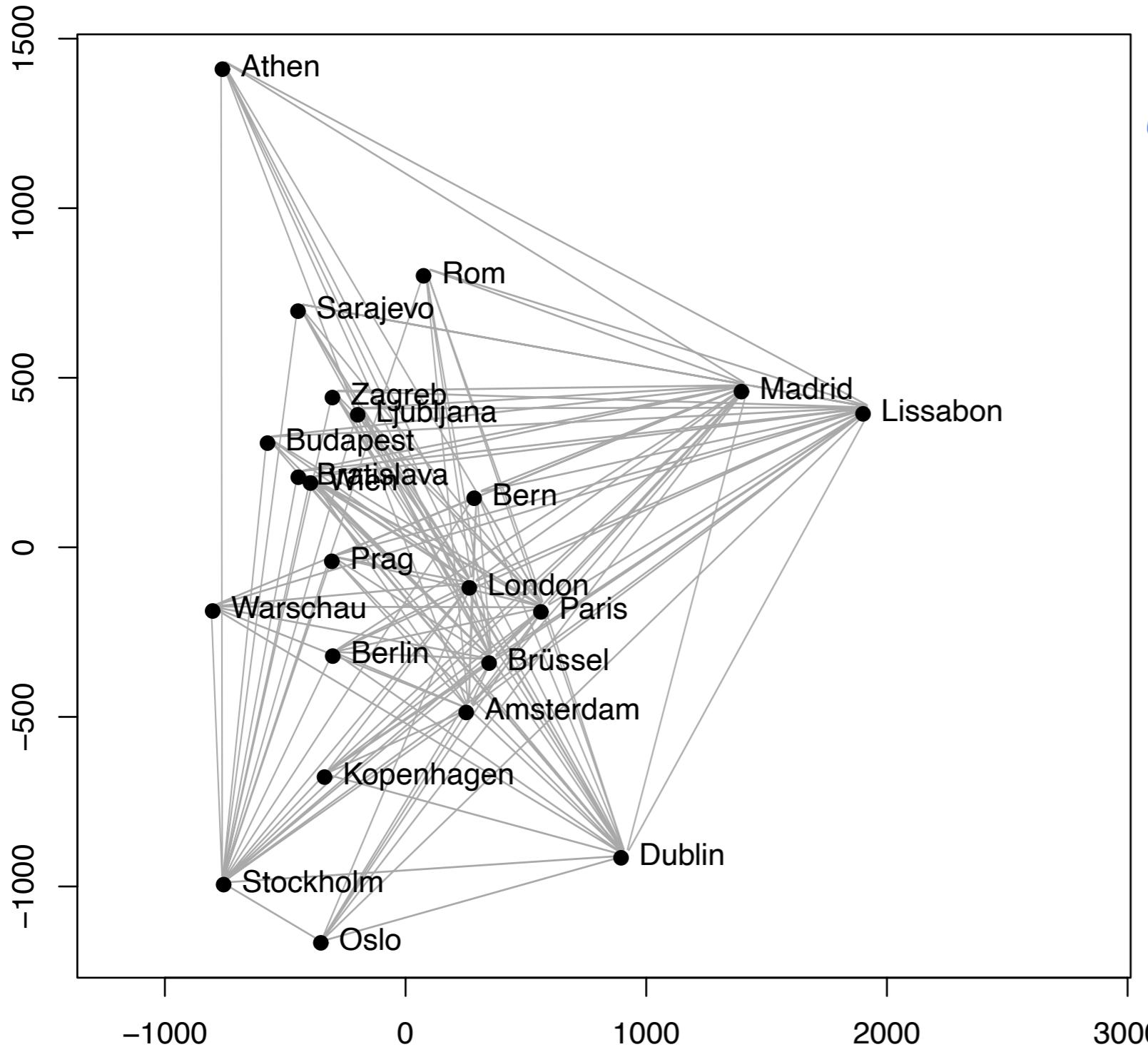
# Multidimensional Scaling

## Hauptstädte Europas (Auswahl)



# Multidimensional Scaling

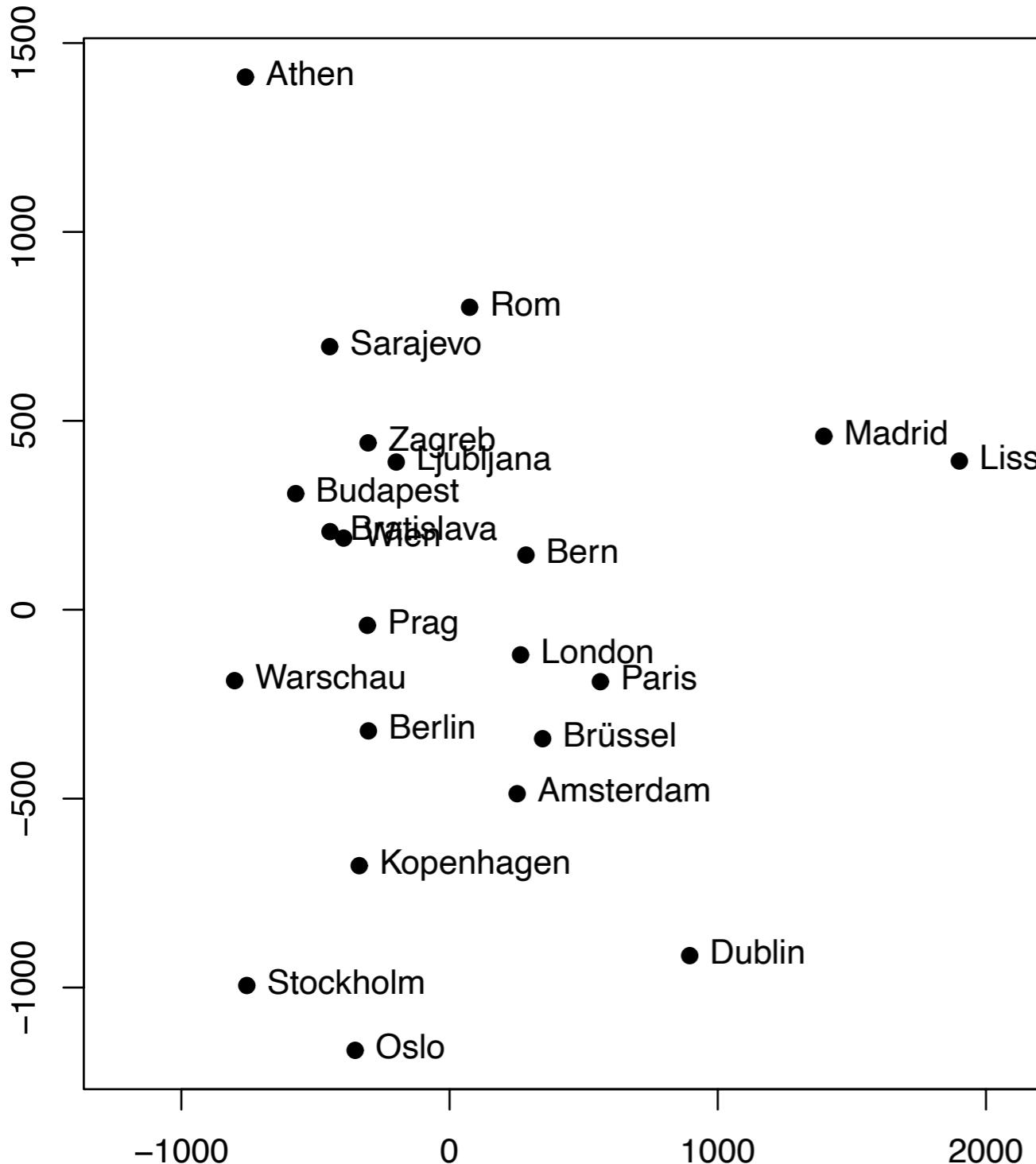
## Hauptstädte Europas (Auswahl)



Graph mit Verbindungen,  
die den Abstände  
entsprechen

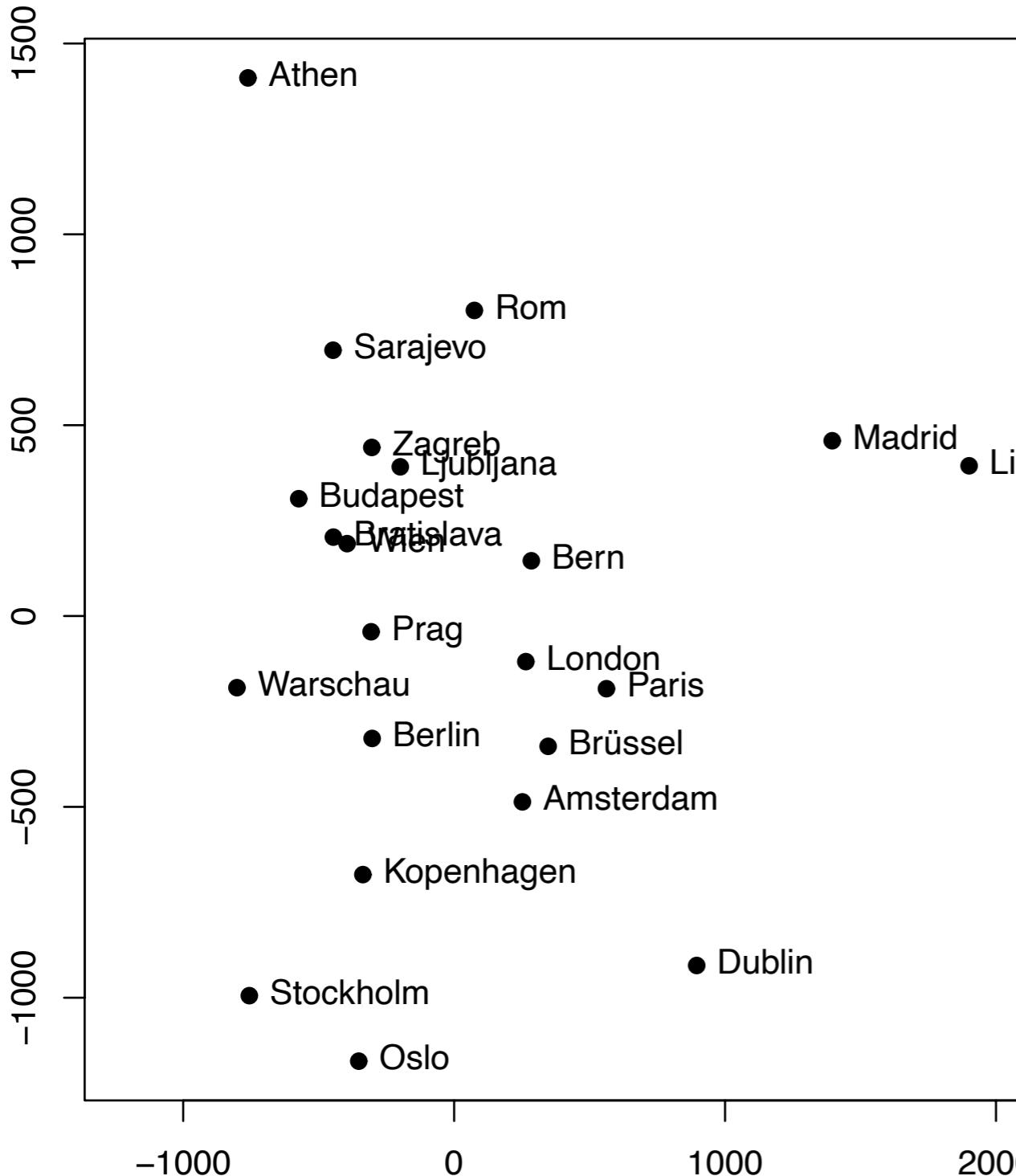
# Multidimensional Scaling

# Hauptstädte Europas (Auswahl)



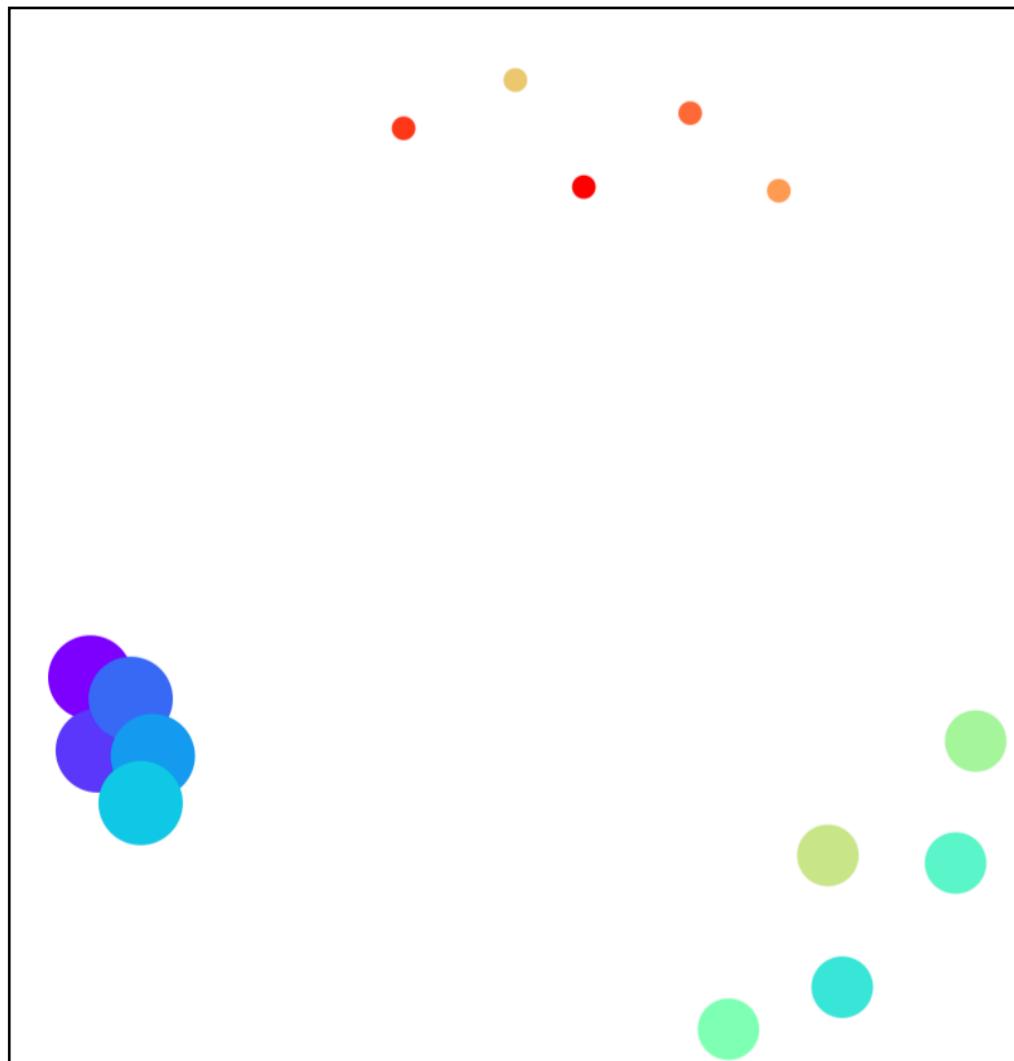
# Multidimensional Scaling

## Hauptstädte Europas (Auswahl)



# Multidimensional Scaling

## Künstliche Daten



# Multidimensional Scaling

MNIST



0 0 0 0 0 0 0 0 0 0 0 0 0 0  
1 1 1 1 1 1 1 1 1 1 1 1 1 1  
2 2 2 2 2 2 2 2 2 2 2 2 2 2  
3 3 3 3 3 3 3 3 3 3 3 3 3 3  
4 4 4 4 4 4 4 4 4 4 4 4 4 4  
5 5 5 5 5 5 5 5 5 5 5 5 5 5  
6 6 6 6 6 6 6 6 6 6 6 6 6 6  
7 7 7 7 7 7 7 7 7 7 7 7 7 7  
8 8 8 8 8 8 8 8 8 8 8 8 8 8  
9 9 9 9 9 9 9 9 9 9 9 9 9 9

hier: 5.000 Samples

⇒ MDS sehr  
rechteckanährend

Bewertung: eher besser als  
PCA (aber trotz. nicht super)

# Multidimensional Scaling

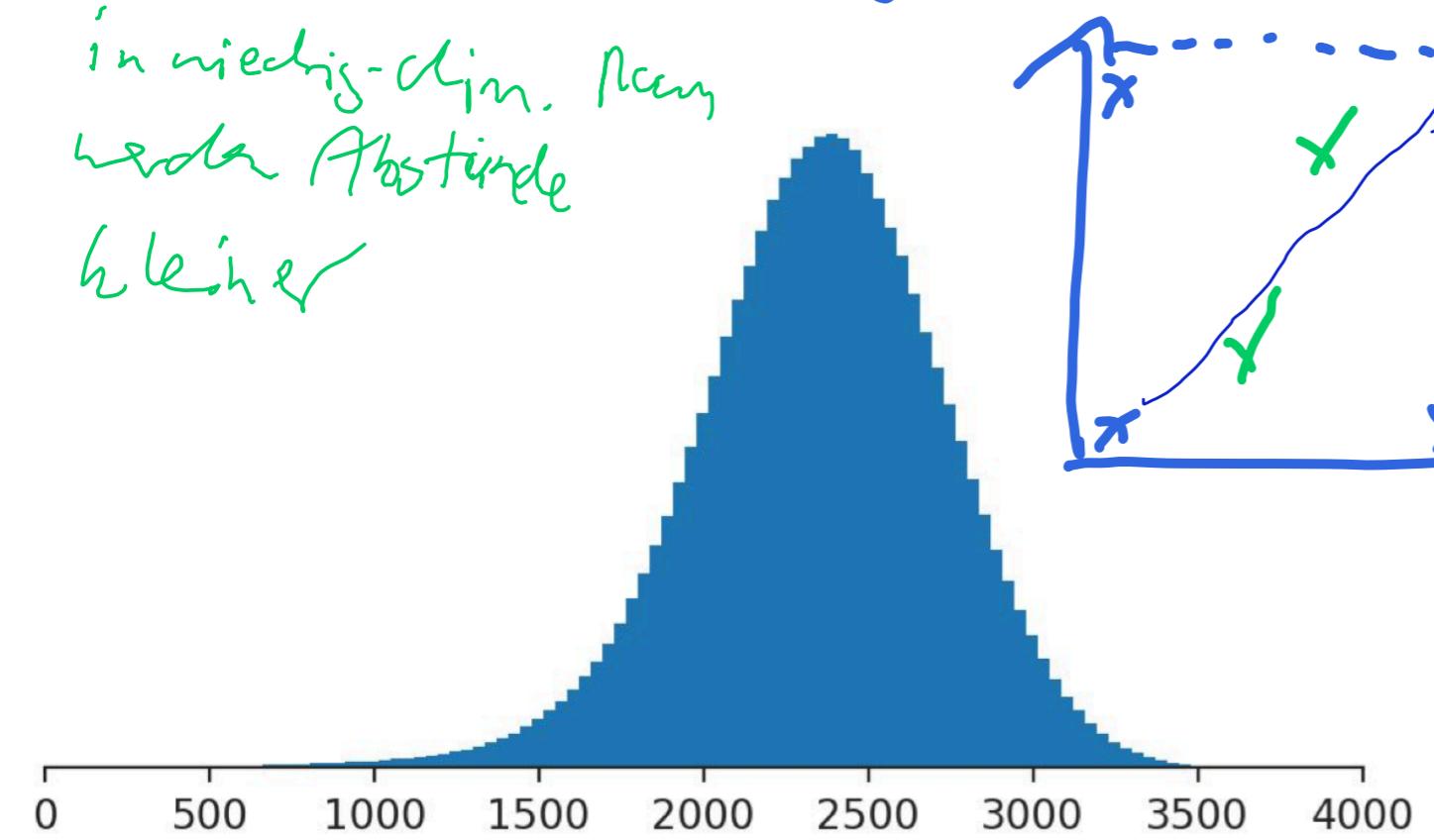
## MNIST

# Curse of Dimensionality

=> hochdim. Raum ist dann  
besetzt

→ Abstände sind groß

in niedrig- oder  
oder Abstand  
kleiner

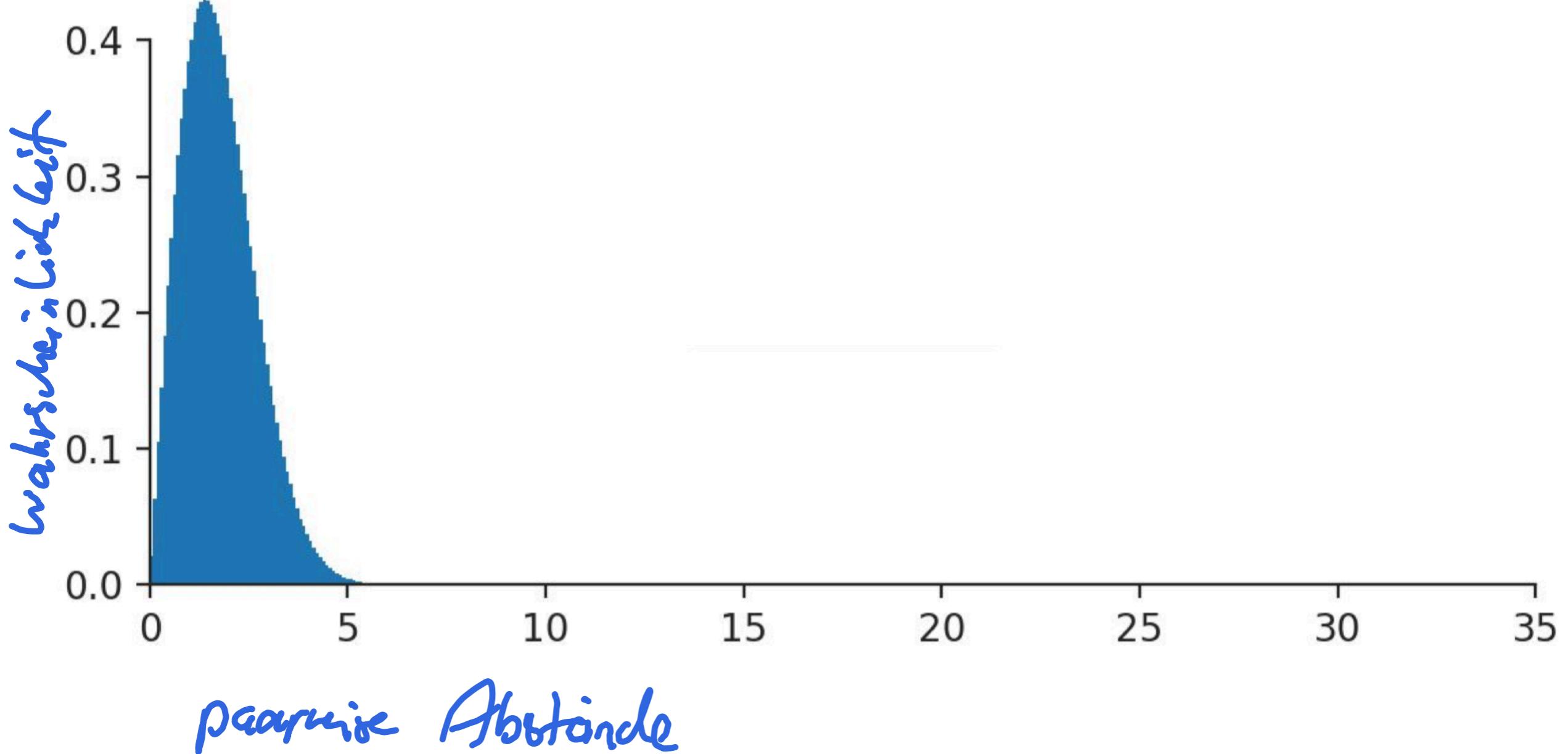


Paarweise Abstände in 784-dim. Raum

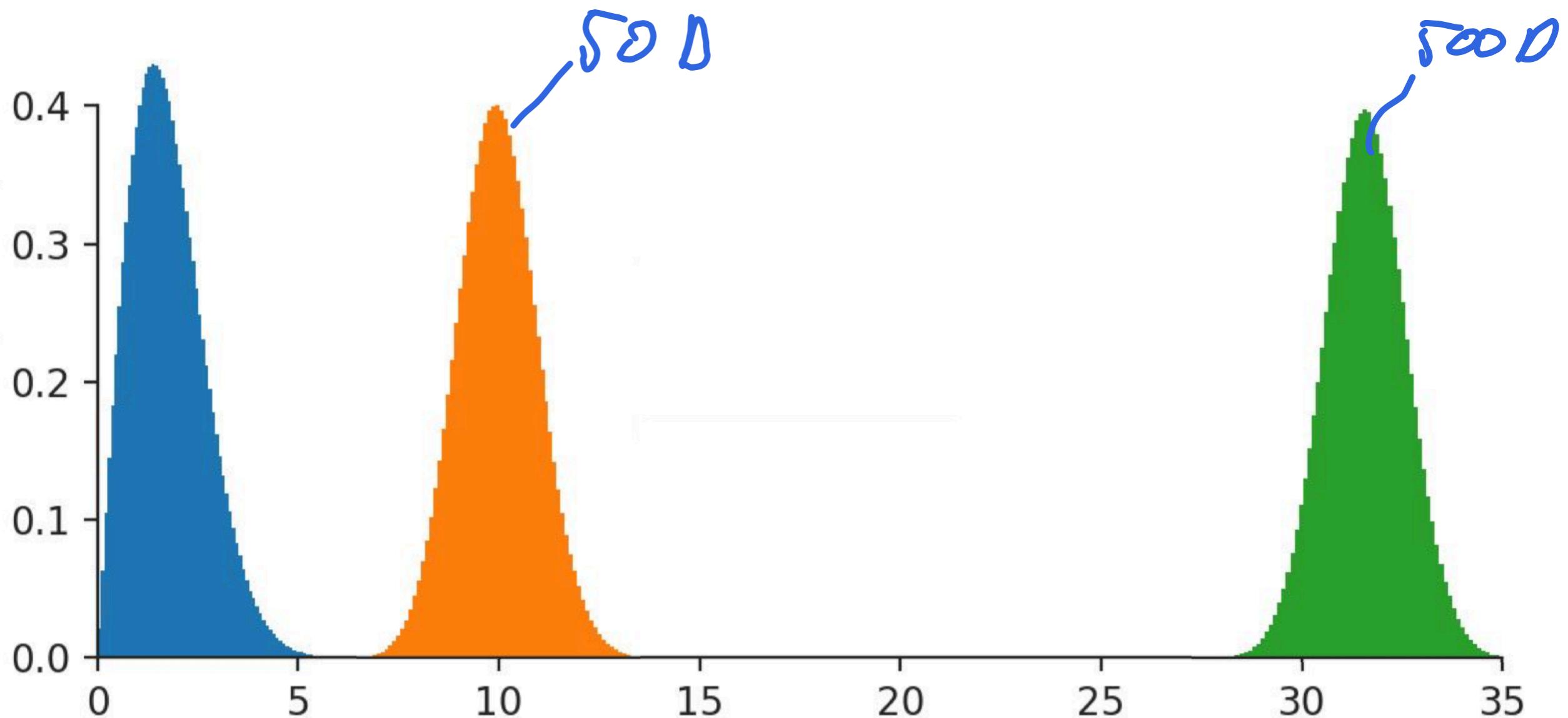


# Multidimensional Scaling

Beispiel: künstliche Daten; gaussverteilt



# Multidimensional Scaling



⇒ MDS kann nicht gut funktionieren, weil große Abstände im niedrig-dimensionalen Raum nicht zu realisieren sind

© D. Kobak, 2021

## [PDF] Stochastic neighbor embedding

[G Hinton, ST Roweis - NIPS, 2002 - Citeseer](#)

We describe a probabilistic approach to the task of placing objects, described by high-dimensional vectors or by pairwise dissimilarities, in a low-dimensional space in a way that preserves neighbor identities. A Gaussian is centered on each object in the high ...

☆ Speichern ⚡ Zitieren Zitiert von: 1715 Ähnliche Artikel Alle 17 Versionen »

7.12.2021

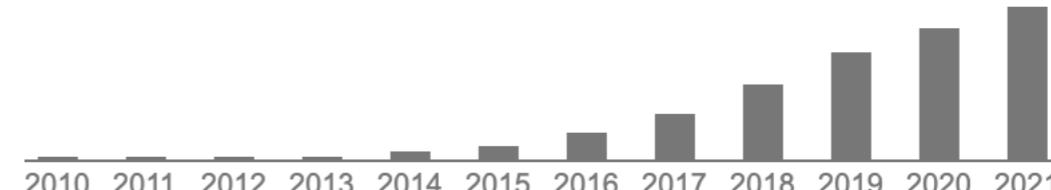
## [PDF] Visualizing data using t-SNE.

[L Van der Maaten, G Hinton - Journal of machine learning research, 2008 - jmlr.org](#)

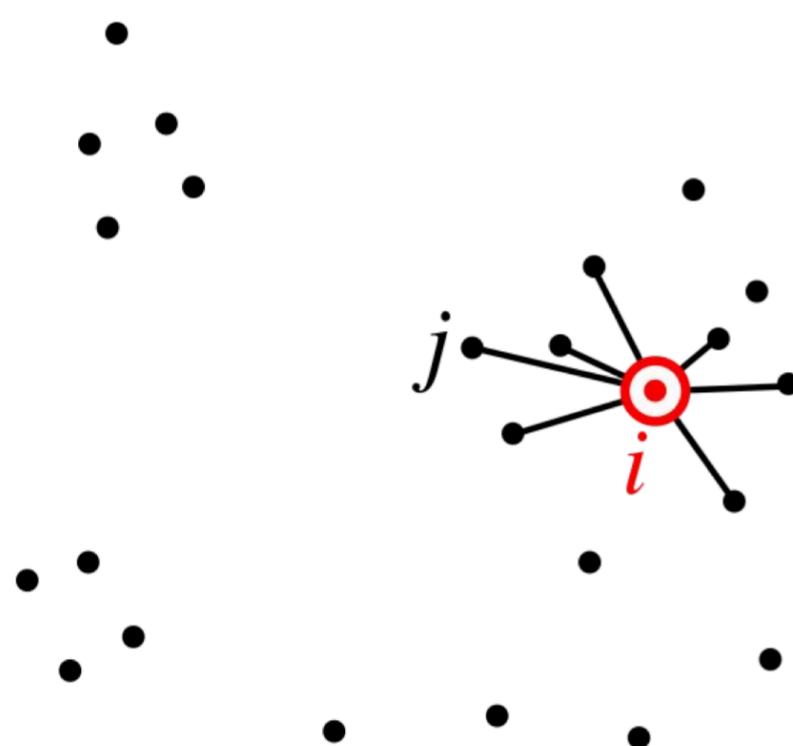
We present a new technique called “t-SNE” that visualizes high-dimensional data by giving each datapoint a location in a two or three-dimensional map. The technique is a variation of Stochastic Neighbor Embedding (Hinton and Roweis, 2002) that is much easier to optimize ...

☆ Speichern ⚡ Zitieren Zitiert von: 23689 Ähnliche Artikel Alle 57 Versionen »

~26300  
heute



# t-SNE



Kalibrier. Leibler-Divergenz

$$D_{\text{t-SNE}} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

$p_{ij}$ : Ähnlichkeit im hochdim. Raum

$q_{ij}$ : Ähnlichkeit im niedrig-dim. Raum

→ Nachbarn im hochdim. Raum sollten wieder Nachbarn  
im niedrig-dim. Raum werden

$p_{ij} \approx q_{ij} \Rightarrow x_i$  und  $x_j$  keine Nachbarn

→  $\log \frac{p_{ij}}{q_{ij}}$  ist negativ, da Multiplikator  $\ll 1$  ( $p_{ij} <$ )

$p_{ij}$  groß

⇒  $x_i$  und  $x_j$  sind nicht Nachbarn

W.Folge:  $q_{ij}$  sollte auch groß sein

negativ:  $\log \frac{p_{ij}}{q_{ij}} \approx 0$

⇒ kleiner Fehler

# t-SNE

gerichtete Ähnlichkeit von  $x_j$  zu  $x_i$

wettf

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

Normalverteilung mit Standardabweichung  $\sigma$   
normiert über alle anderen Abstände zu  $x_i$

$\sigma$  steckt  
welche Nachbarschaften  
erfasst werden

⇒ weit entfernte Nachbarn haben eine gerichtete Ähnlichkeit von Nähe 0

$P = 2^H$  - Perplexität ( $= 2^{H_{\text{entropic}}}$ ): je breiter die Normalverteilung, desto höher die Perplexität

$$H_{\text{entropic}} = - \sum_{j \neq i} p_{j|i} \log p_{j|i}$$

⇒ Anzahl einbezogener Nachbarn

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

ganz auch viel einfacher

# t-SNE

man kann auch approximieren ~ fragt er nicht

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

$p_{j|i} \rightarrow \frac{1}{k}$  für  $k$  nächste Nachbarn

Annahme  
⇒ Gleichverteilung      1, Sich  $k$  nächste Nachbarn in hochdim. Raum  
2, Vertrele Ähnlichkeit gleichmäßig

$$\mathcal{P} = 2^H$$
$$H = - \sum_{j \neq i} p_{j|i} \log p_{j|i}$$

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

# t-SNE

$$q_{ij} = \frac{w_{ij}}{Z}$$

$$Z = \sum_{k \neq l} w_{kl}$$

$$w_{ij} = k(d)$$

$$d = \|z_i - z_j\|$$

↳ Ähnlichkeitskernel

↳  $k(d) = \exp(-d^2)$

Normalverteilung

→ SNE

↳  $k(d) = \frac{1}{1 + d^2}$

Laplace-Verteilung

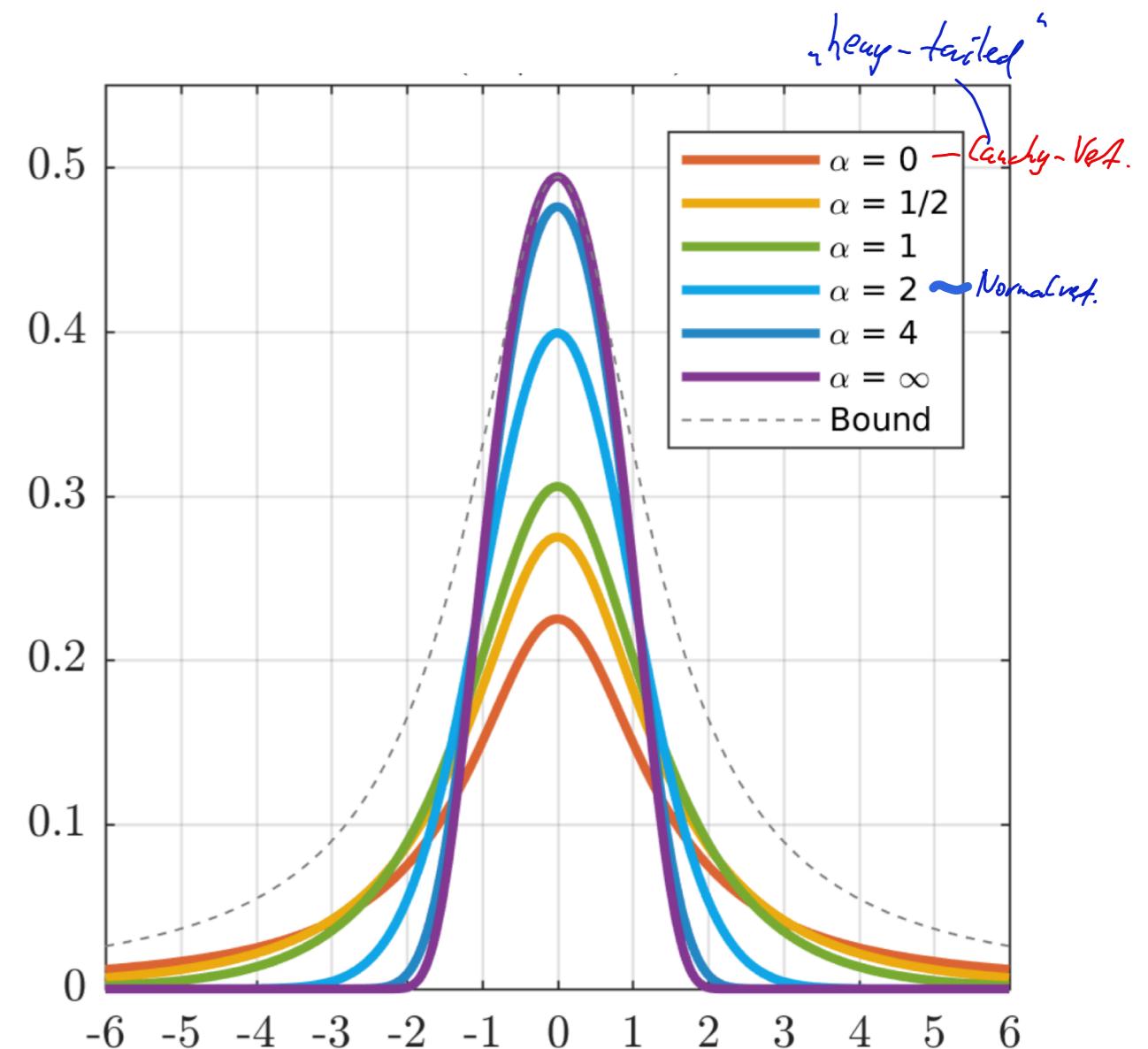
→ t-SNE

# t-SNE

Cauchy-Verteilung:  
flacher Abfall zum  
Rand hin

$$k(d) = \exp(-d^2)$$

$$k(d) = \frac{1}{1 + d^2}$$



# t-SNE

## Training

Gradientenabstieg

Initialisierung: Zufällige Punkte im niedrigdimensionalen Raum

$$D_{\text{t-SNE}} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \geq \sum_{i,j} p_{ij} \log p_{ij} - \sum_{i,j} p_{ij} \log q_{ij}$$

nur  $q$  kann beeinflusst  
werden

bleibt unverändert  
 $\rightarrow$  nicht berücksichtigt

$$\geq - \sum_{i,j} p_{ij} \log \frac{w_{ij}}{z} = - \sum_{i,j} p_{ij} \log w_{ij} + \sum_{i,j} p_{ij} \log z$$

$$\geq - \sum_{i,j} p_{ij} \log w_{ij} + \sum_{i,j} p_{ij} \log \sum_{k,l} w_{kl}$$

$$= - \sum_{i,j} p_{ij} \log w_{ij} + \log \sum_{k,l} w_{kl} \cdot \sum_{i,j} p_{ij} \Rightarrow = - \sum_{i,j} p_{ij} \log q_{ij} + \log \sum_{i,j} w_{ij}$$

# t-SNE

## Training

$w_{ij}$  ist Abstand ( $r_{ij}$ ) durchgeschoben durch Cauchy-Verteilung  
→ Ergebnis ist groß wenn  $\tau$  klein ist

Gradientenabstieg

Initialisierung: Zufällige Punkte im niedrigdimensionalen Raum

$$D_{\text{t-SNE}} = - \sum_{i,j} p_{ij} \log \frac{w_{ij}}{q_{ij}} + \log \sum_{i,j} w_{i,j}$$

$\underbrace{\quad}_{1. \text{ Term}}$   $\underbrace{\quad}_{2. \text{ Term}}$

1. Term: soll möglichst groß werden

⇒  $q_{ij}$  groß

⇒  $w_{ij}$  groß ⇒ kleiner Abstand von  $z_i$  und  $z_j$

2. Term: soll klein werden ⇒  $w_{ij}$  klein

⇒ abstoßende Kräfte zwischen  $z_i$  und  $z_j$

## Training

Gradientenabstieg

Initialisierung: Zufällige Punkte im niedrigdimensionalen Raum

$$D_{\text{t-SNE}} = - \sum_{i,j} p_{ij} \log \frac{w_{ij}}{Z}$$

$$\frac{\partial D_{\text{t-SNE}}}{\partial z_i} = -2 \sum_j p_{ij} \frac{1}{w_{ij}} + 2 \frac{1}{Z} \sum_j \frac{\partial w_{ij}}{\partial z_i}$$

$$\sim \sum_j p_{ij} w_{ij} (z_i - z_j) - \frac{1}{Z} \sum_j w_{ij}^2 (z_i - z_j)$$

→ sehr rechenintensiv  $O(n^2)$  je ein abstandsmaß für jedes Paar von Punkten zu berechnen und  $O(n)$

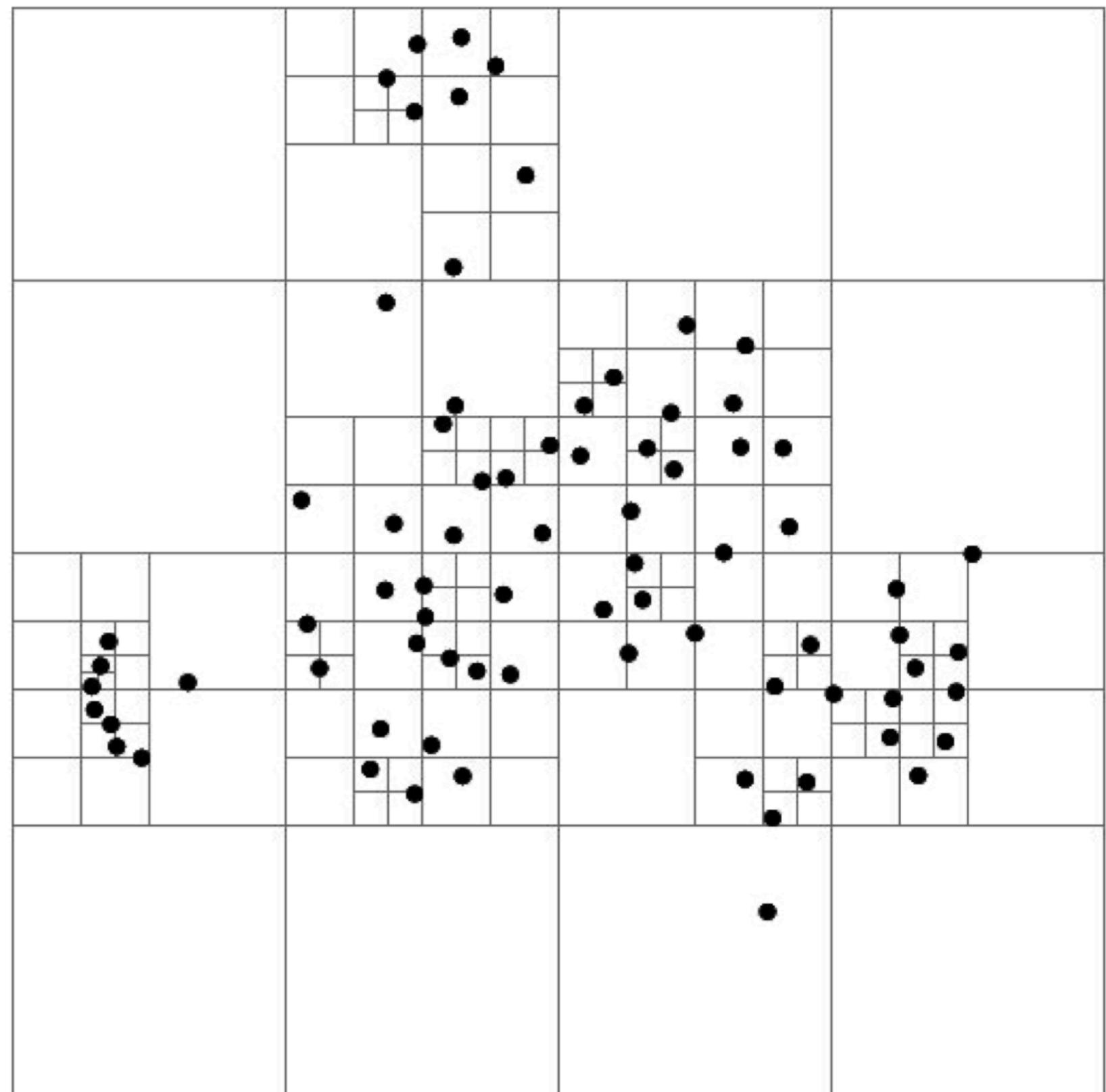
# t-SNE

Barnes-Hut

$\hookrightarrow$  Approximation von k-NN-Verfahren

$\Rightarrow$  95% sind tatsächlich  
nächste Nachbarn, 5% nicht

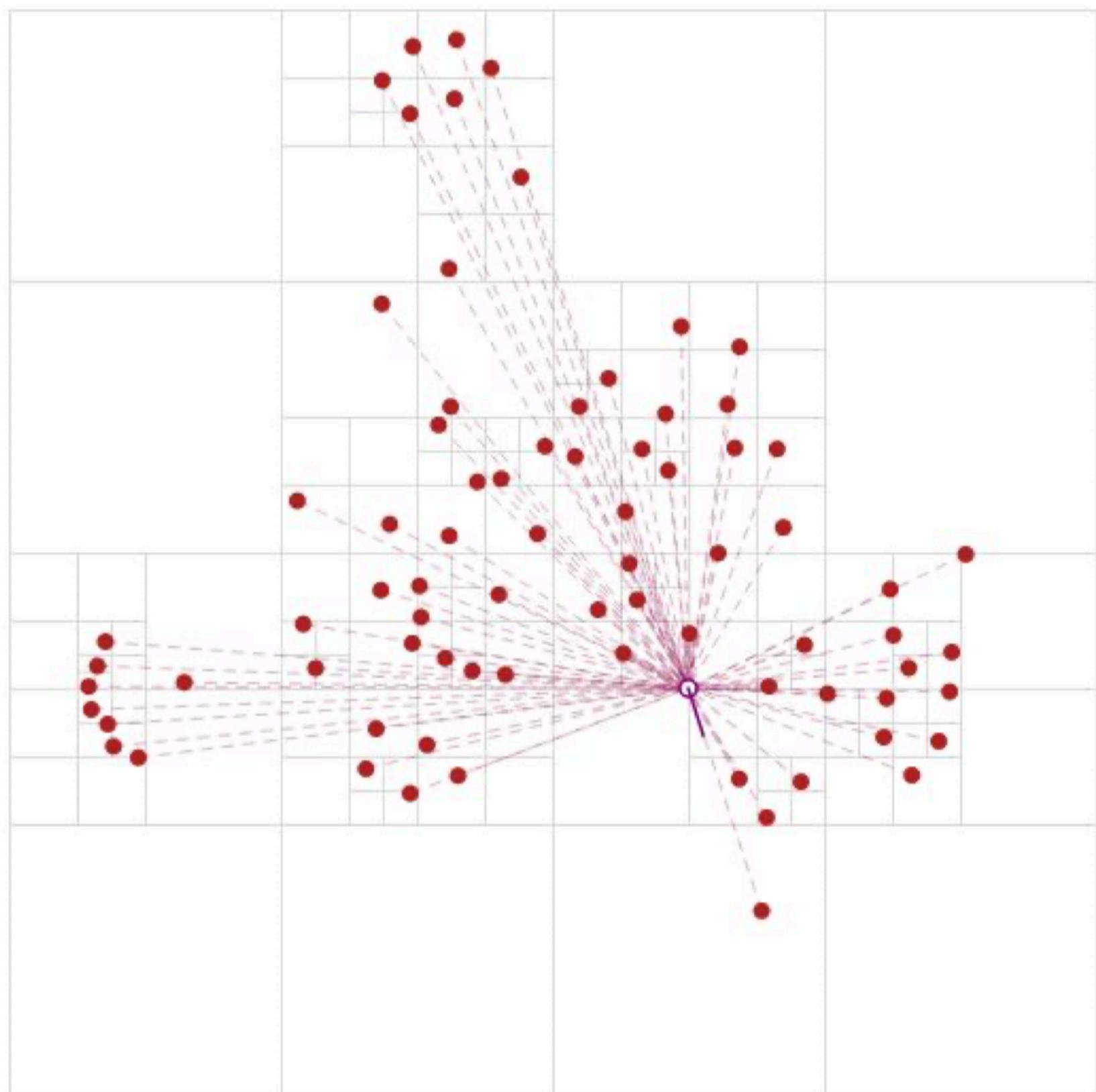
$\Rightarrow O(n \log n)$



© D. Kobak, 2021

# t-SNE

## Barnes-Hut



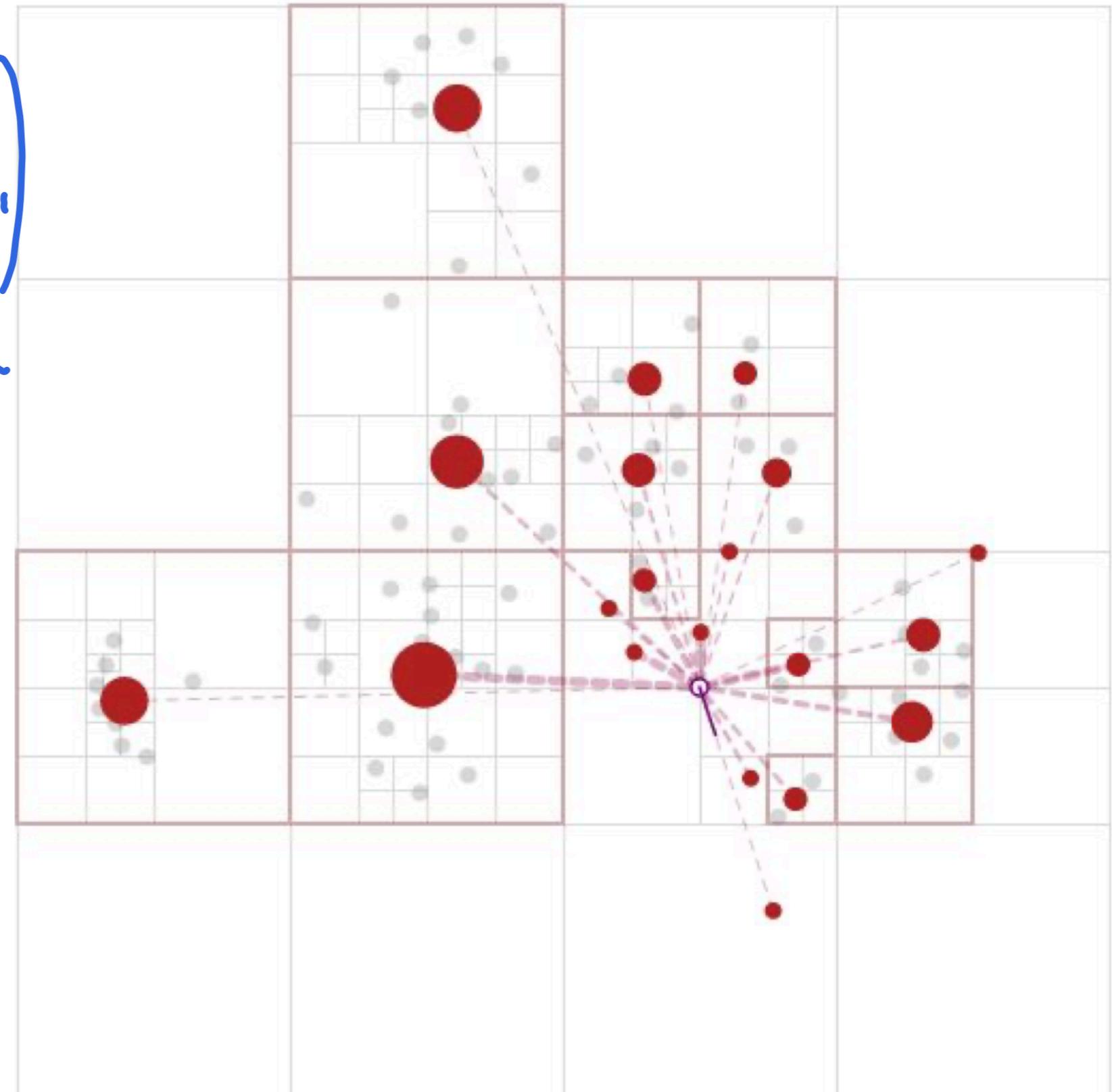
© D. Kobak, 2021

# t-SNE

## Barnes-Hut

Idee: Punkte zusammenfassen;  
Größe des Punktes abhängig davon,  
wie viele Nachbarn enthalten sind

Idee: Simplex-Punkte, die weit  
weg sind zusammenfassen

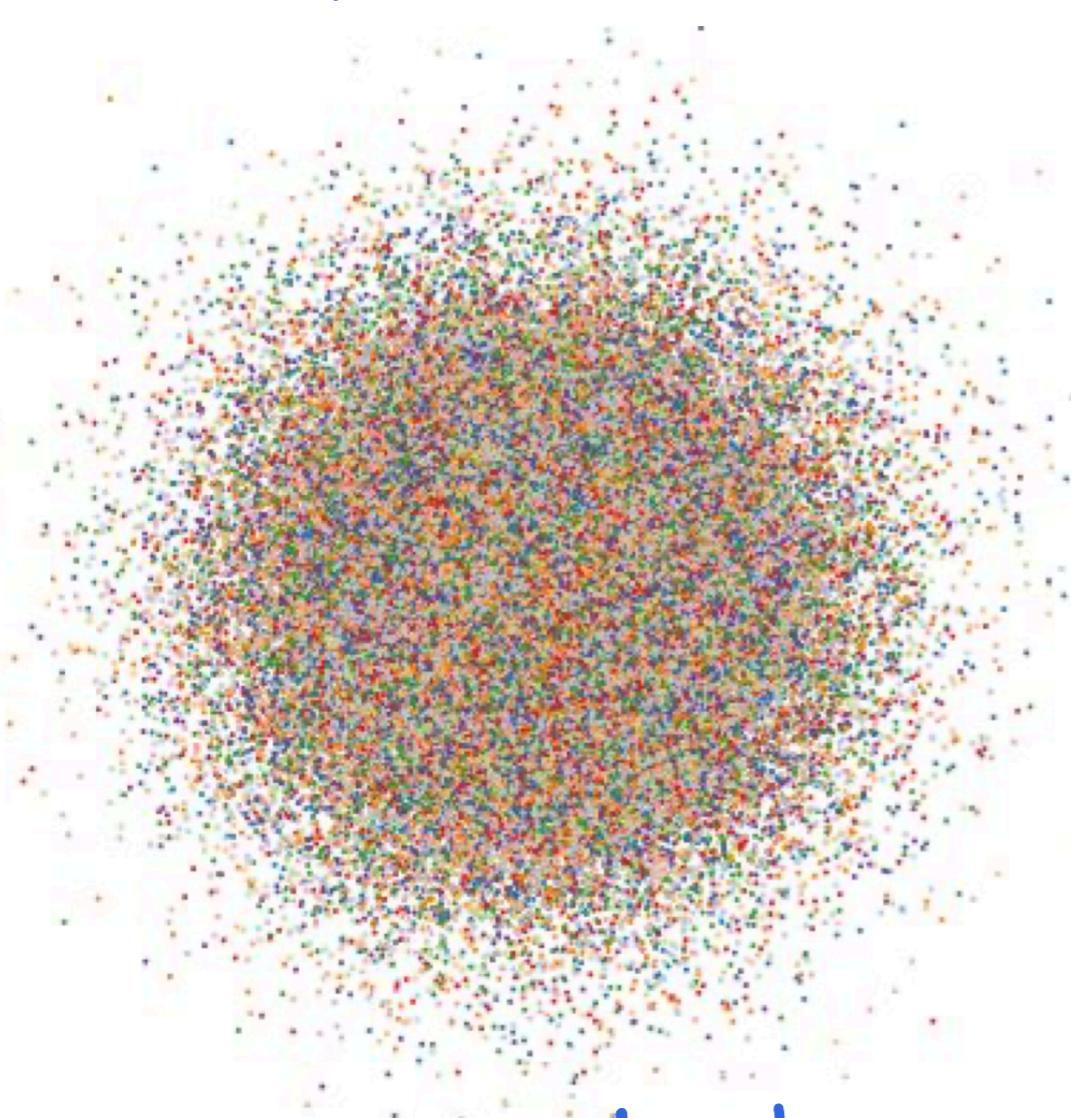


© D. Kobak, 2021

# t-SNE

Initialisierung

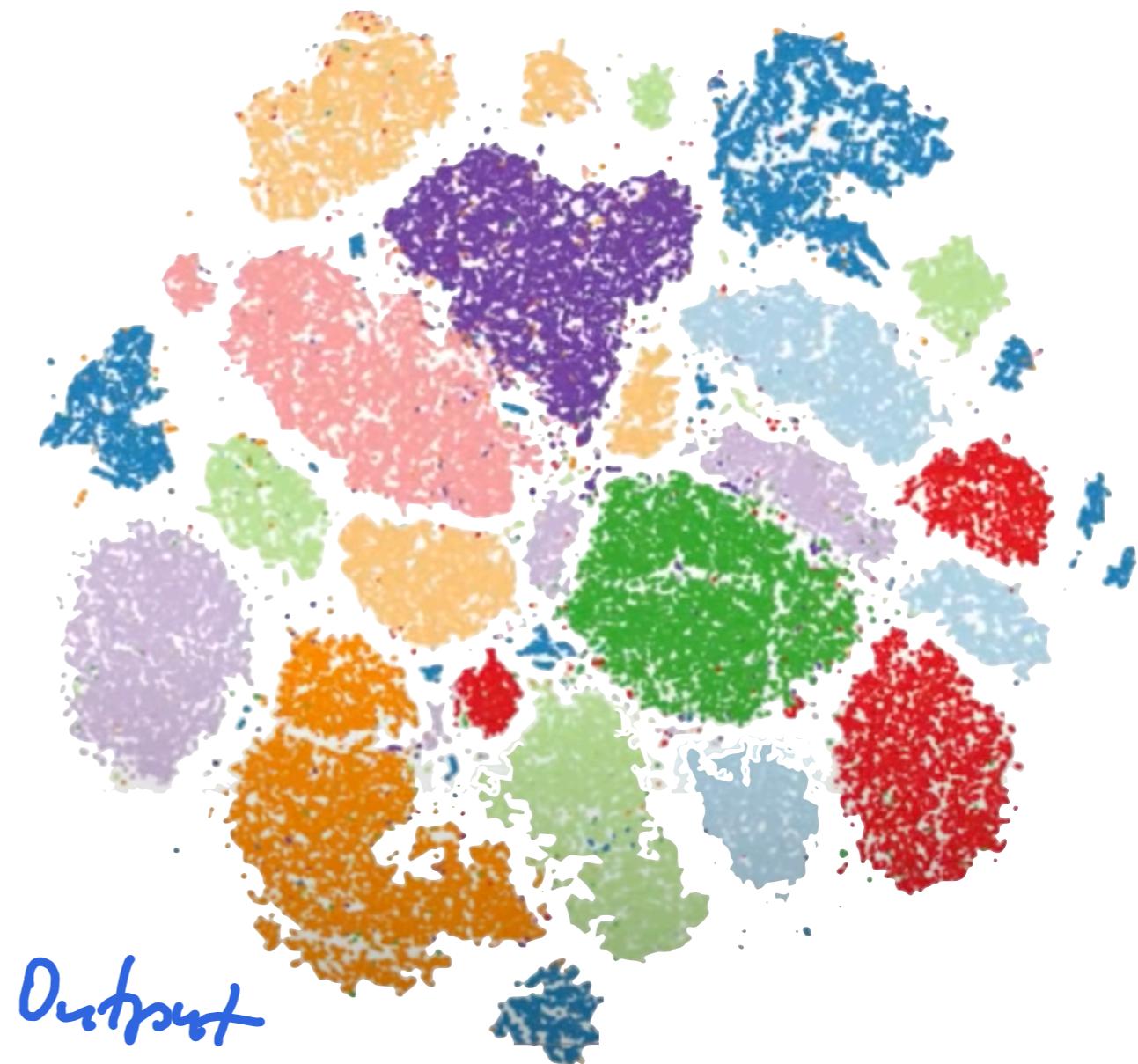
↳ Klassenzugehörigkeit farbig kodiert



Input

DSP : MNIST

t-SNE Originalversion



Output

Perplexity 30

© D. Kobak, 2021

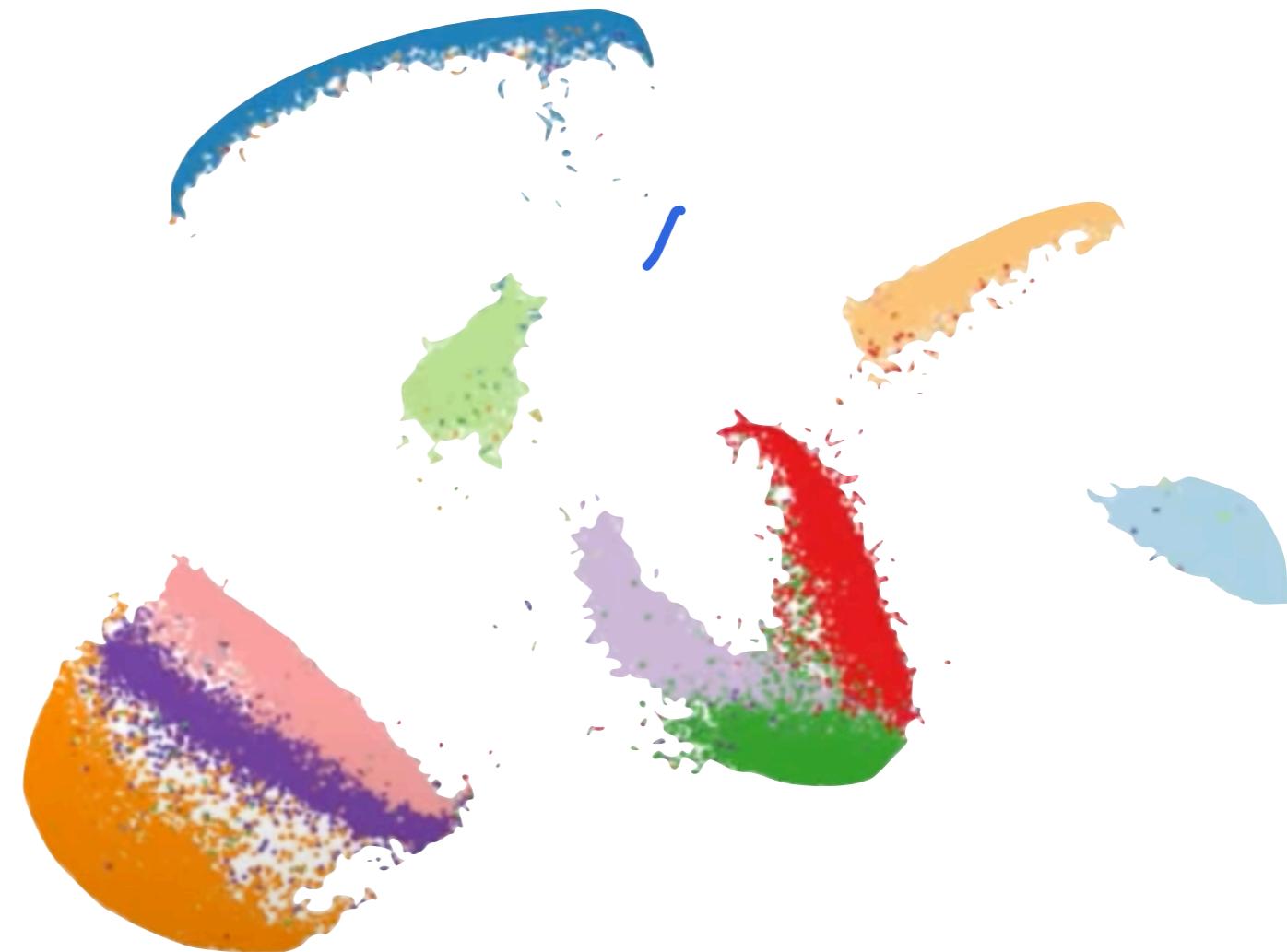
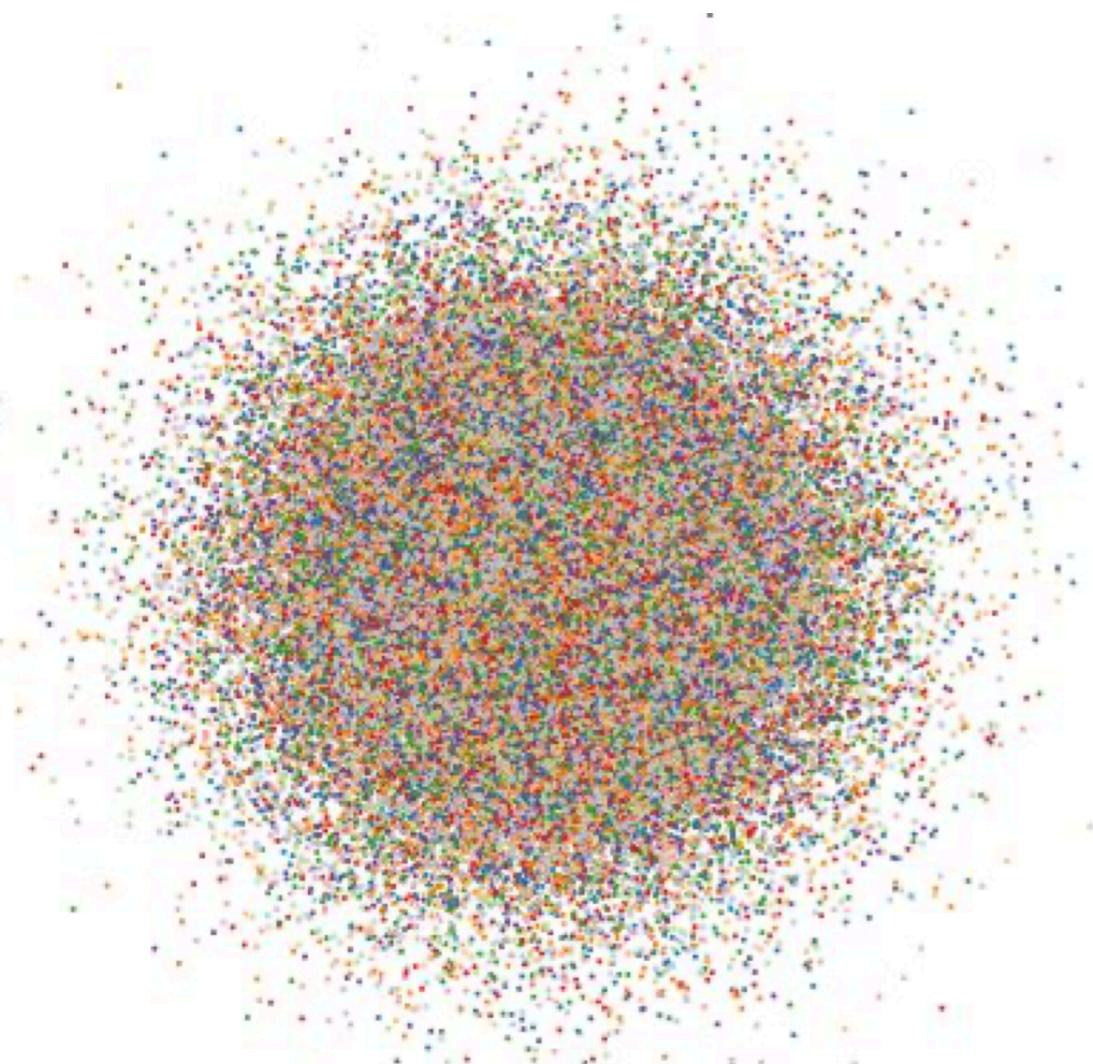
- ↳ gute Clustererkennung
- ↳ Cluster und Klassen passen nicht in allen Fällen zusammen

⇒ Klasse in verschiedene Cluster aufteilen

# t-SNE

Early Exaggeration

= in den ersten Iterationen die anziehenden Kräfte verstärken

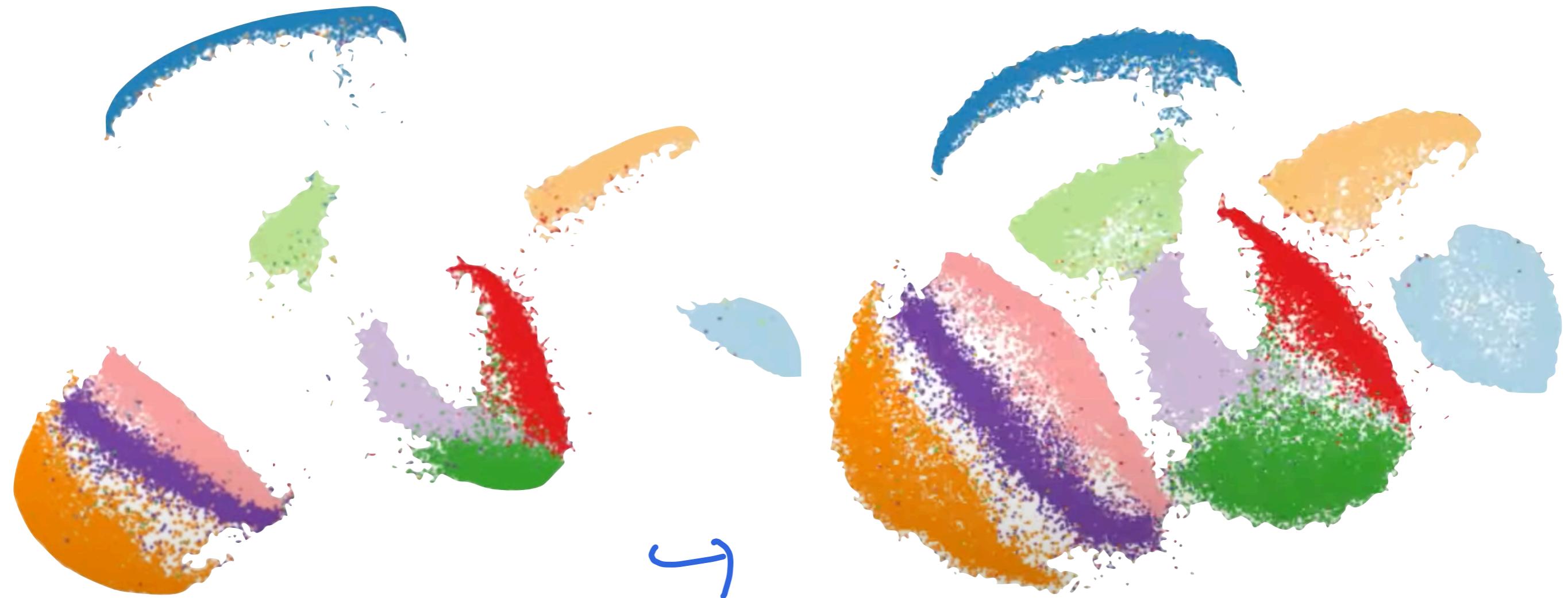


hier: Multiplikator 0 mit 12  
für 250 Iterationen

© D. Kobak, 2021

# t-SNE

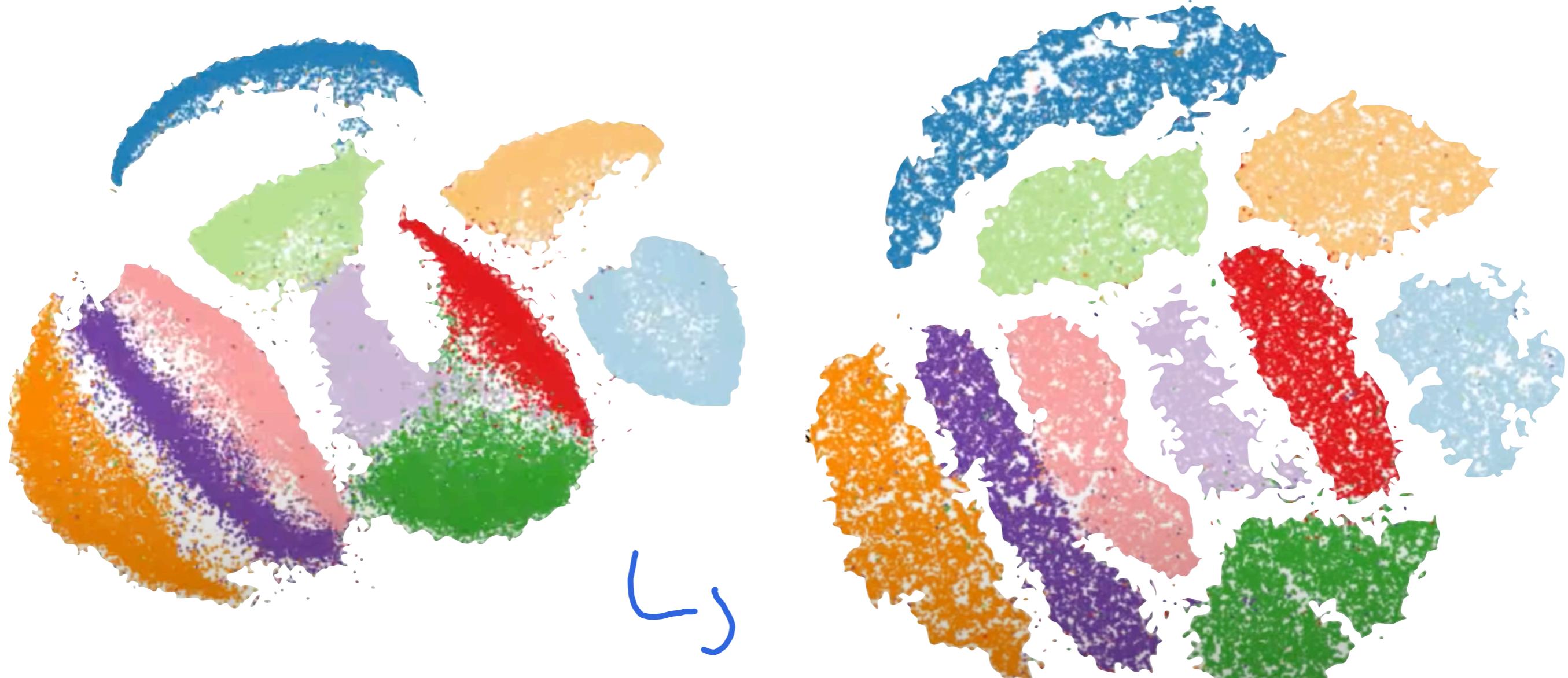
## Early Exaggeration



© D. Kobak, 2021

# t-SNE

## Early Exaggeration



Endesgebnis  $\rightarrow$  gute Klassentrennung

© D. Kobak, 2021

# t-SNE

## Perplexität



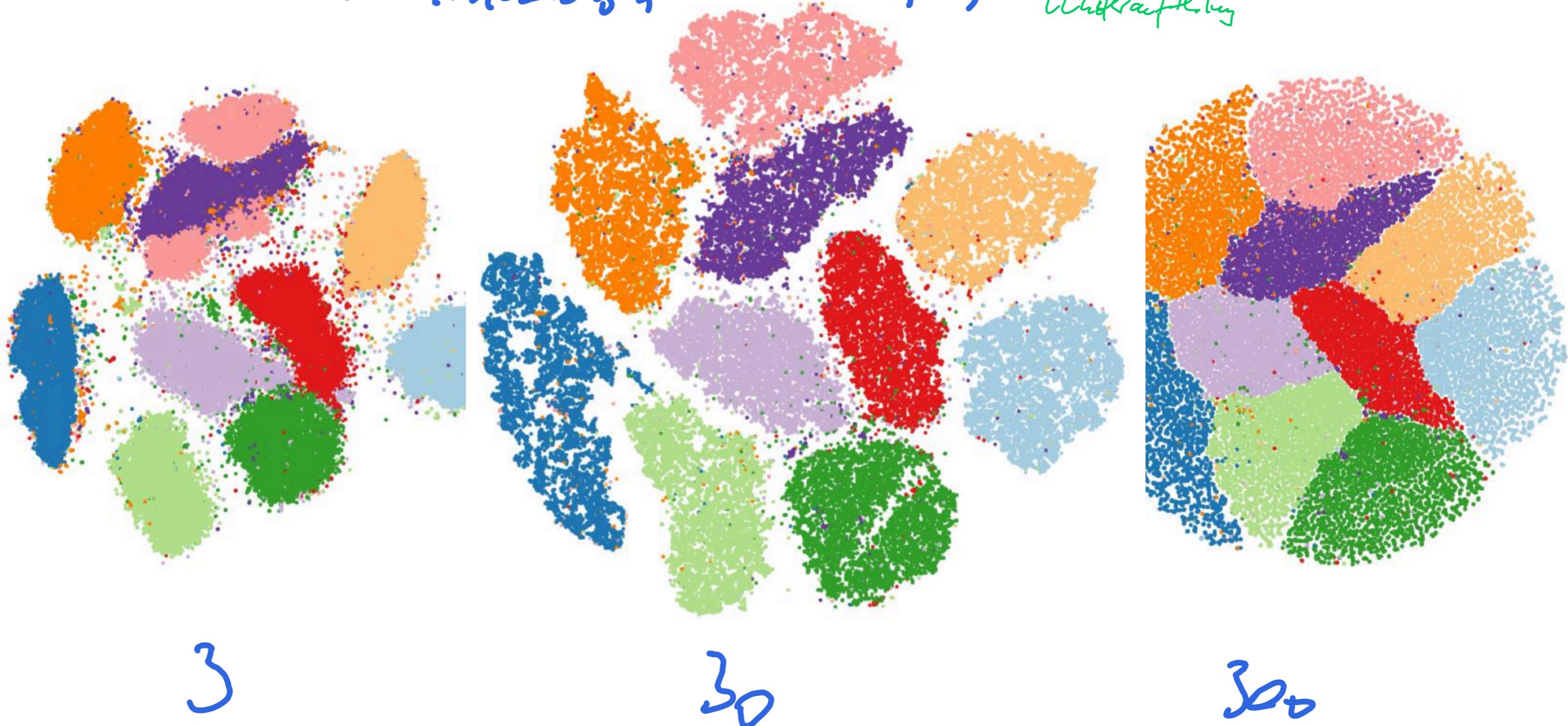
# t-SNE

Perplexität

zu hohe Perplexität führt zu

Zerstreuung zwischen den Clustern

zu niedrige führt zu  
Clusterfusion

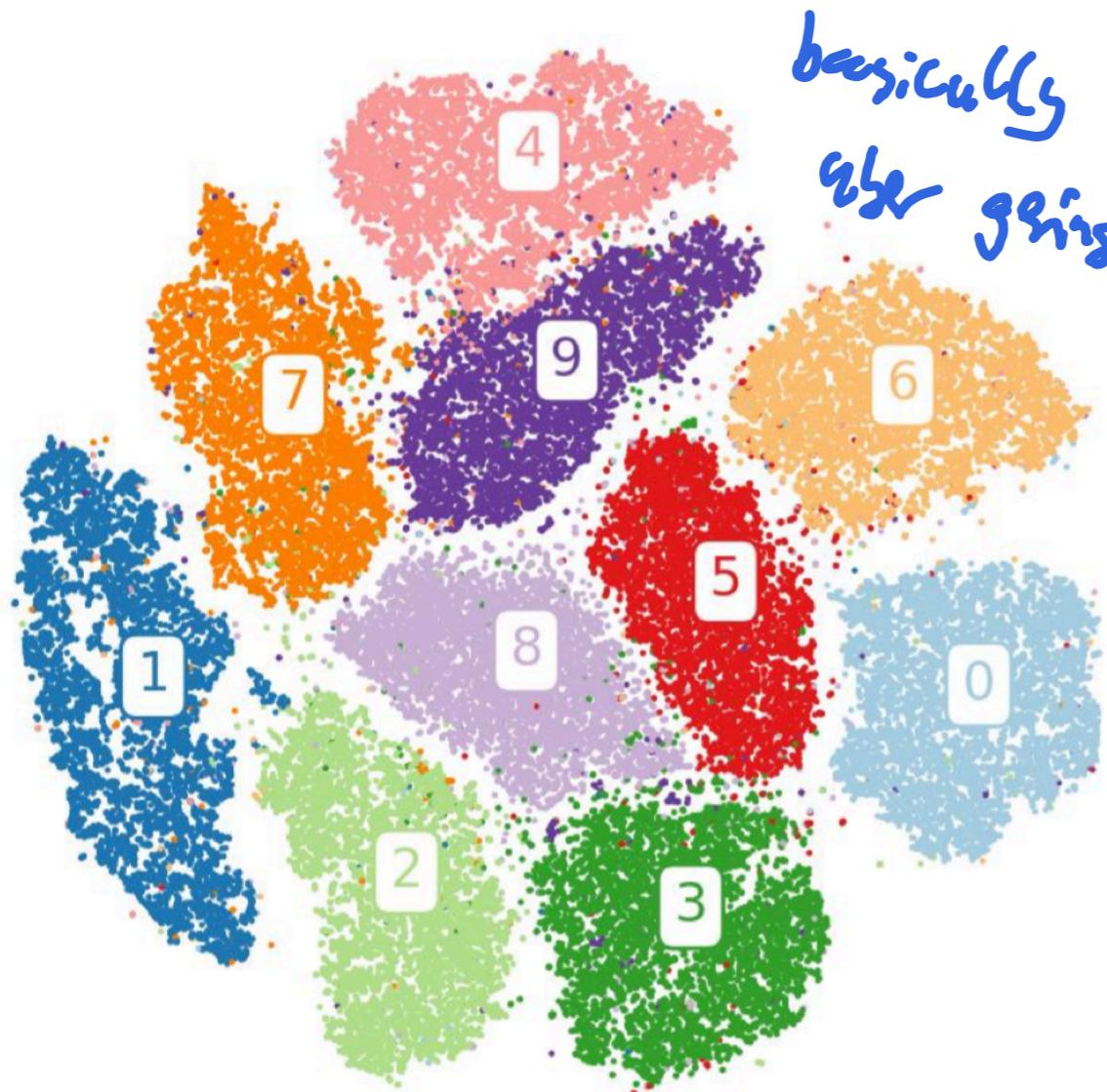


© D. Kobak, 2021

# t-SNE

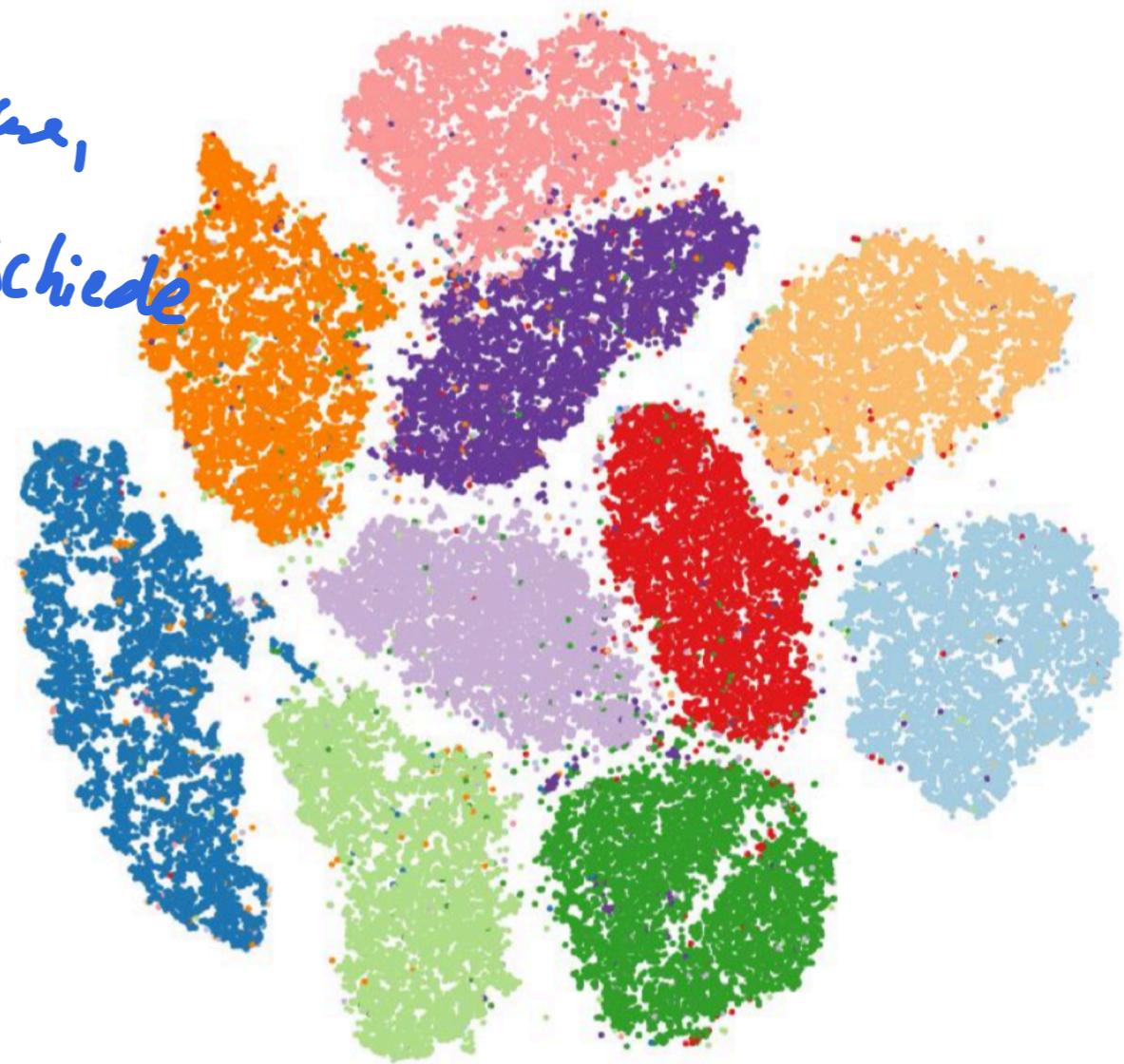
Normalverteilter Kernel vs. gleichverteilter Kernel

Trisine vs. tProportion



normalverteilt

basically the same,  
aber geringe Unterschiede



gleichverteilt

# t-SNE

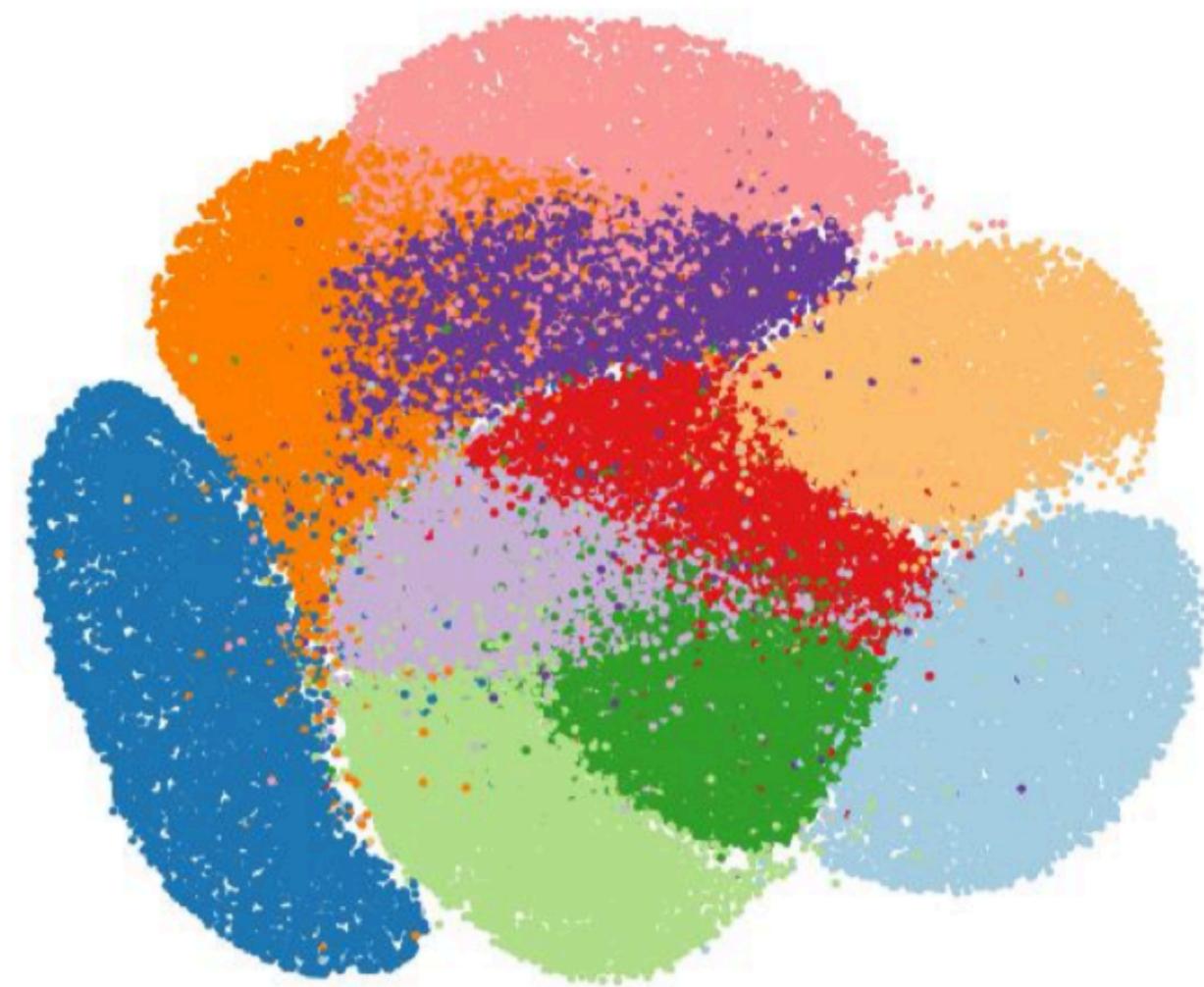
SNE vs. t-SNE

1) *heraltschaft*

2) *bessere Clusterbelegung*



*t-SNE*



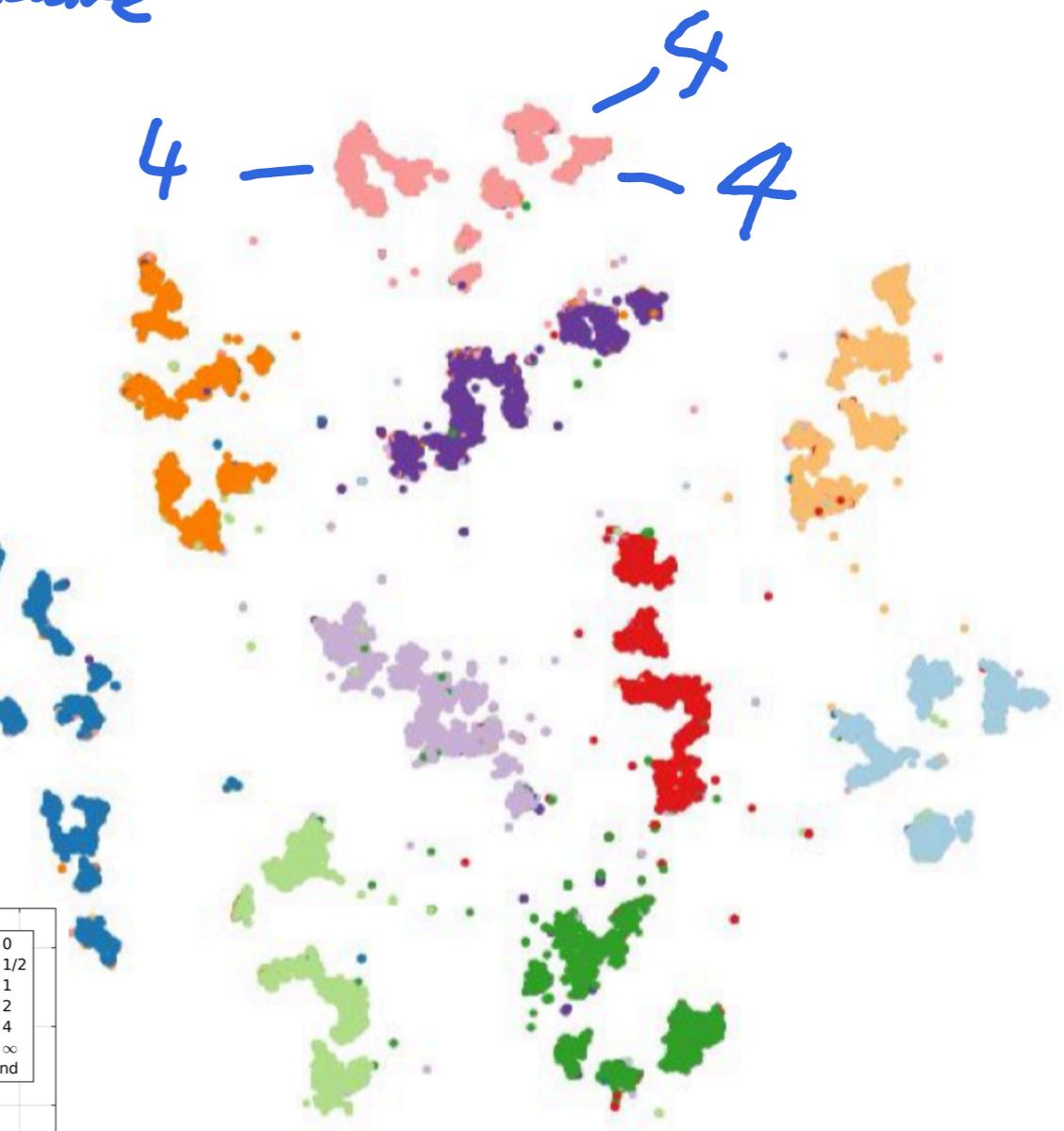
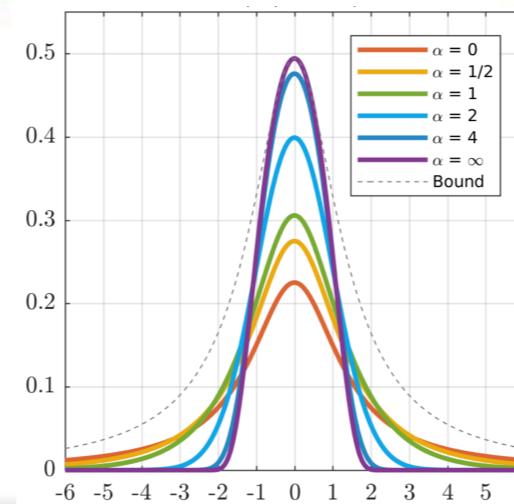
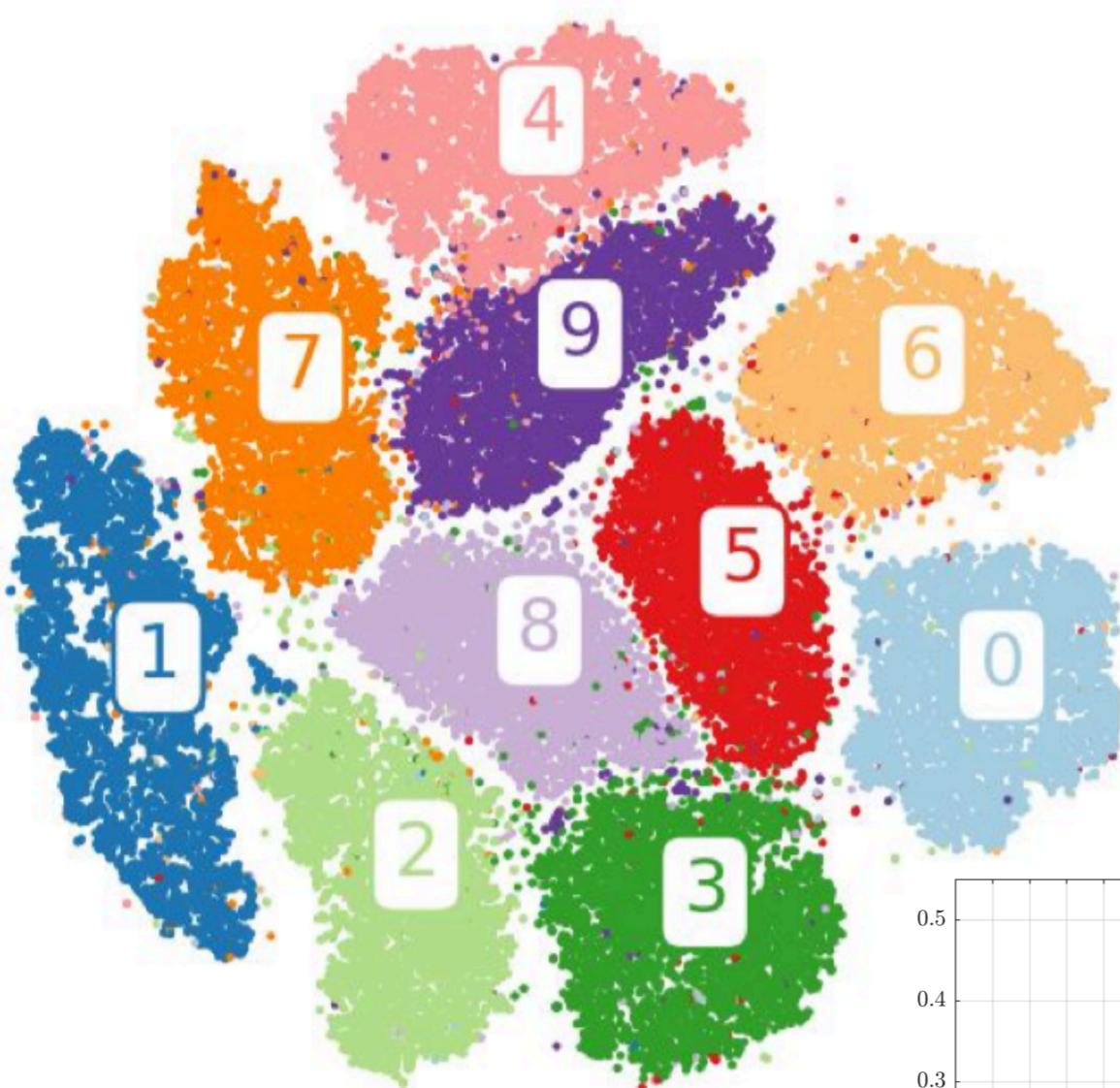
*SNE*

© D. Kobak, 2021

# t-SNE

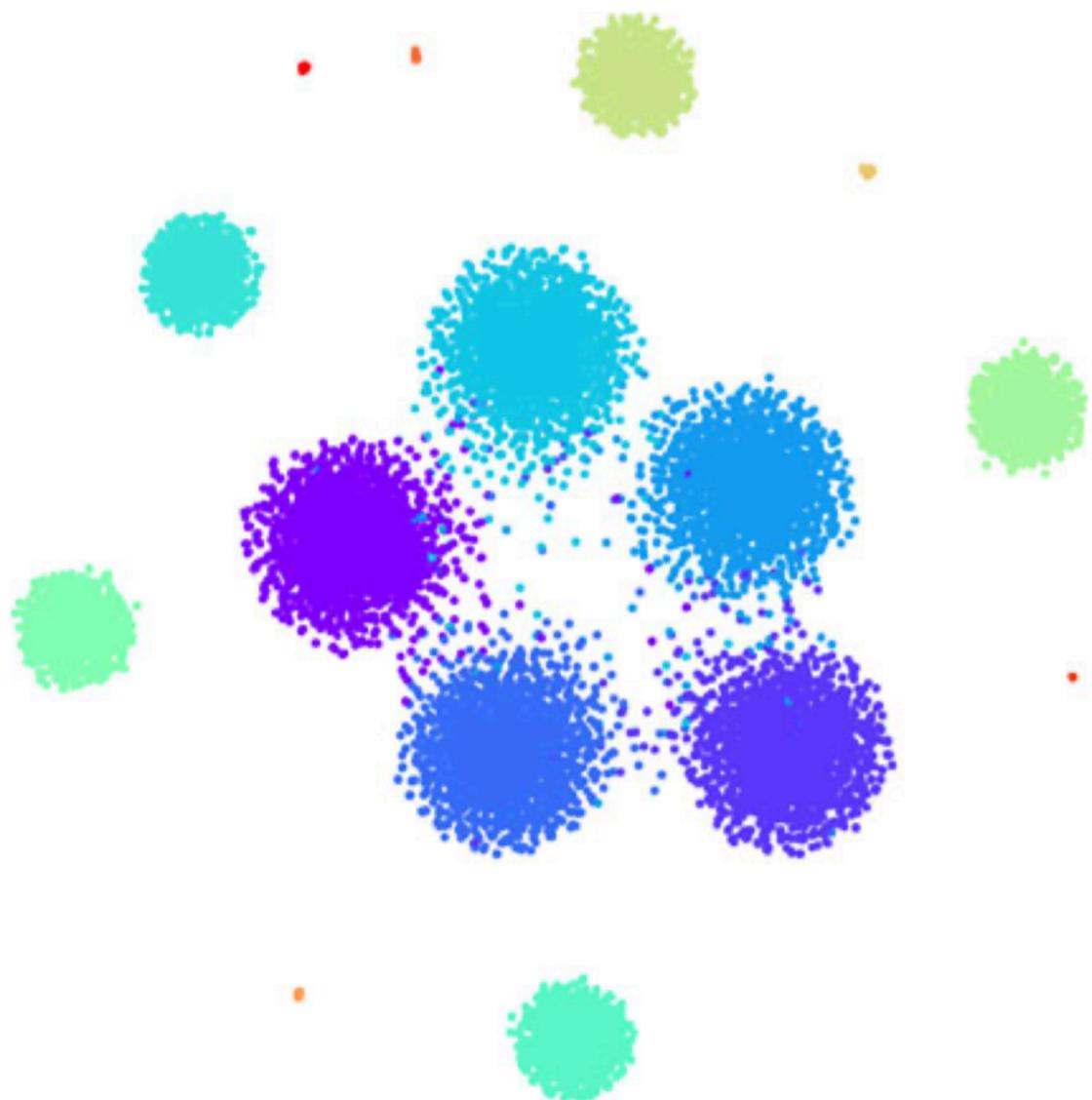
SNE vs. t-SNE

↳ nach flacherer Abfall als  
Kernfunktion



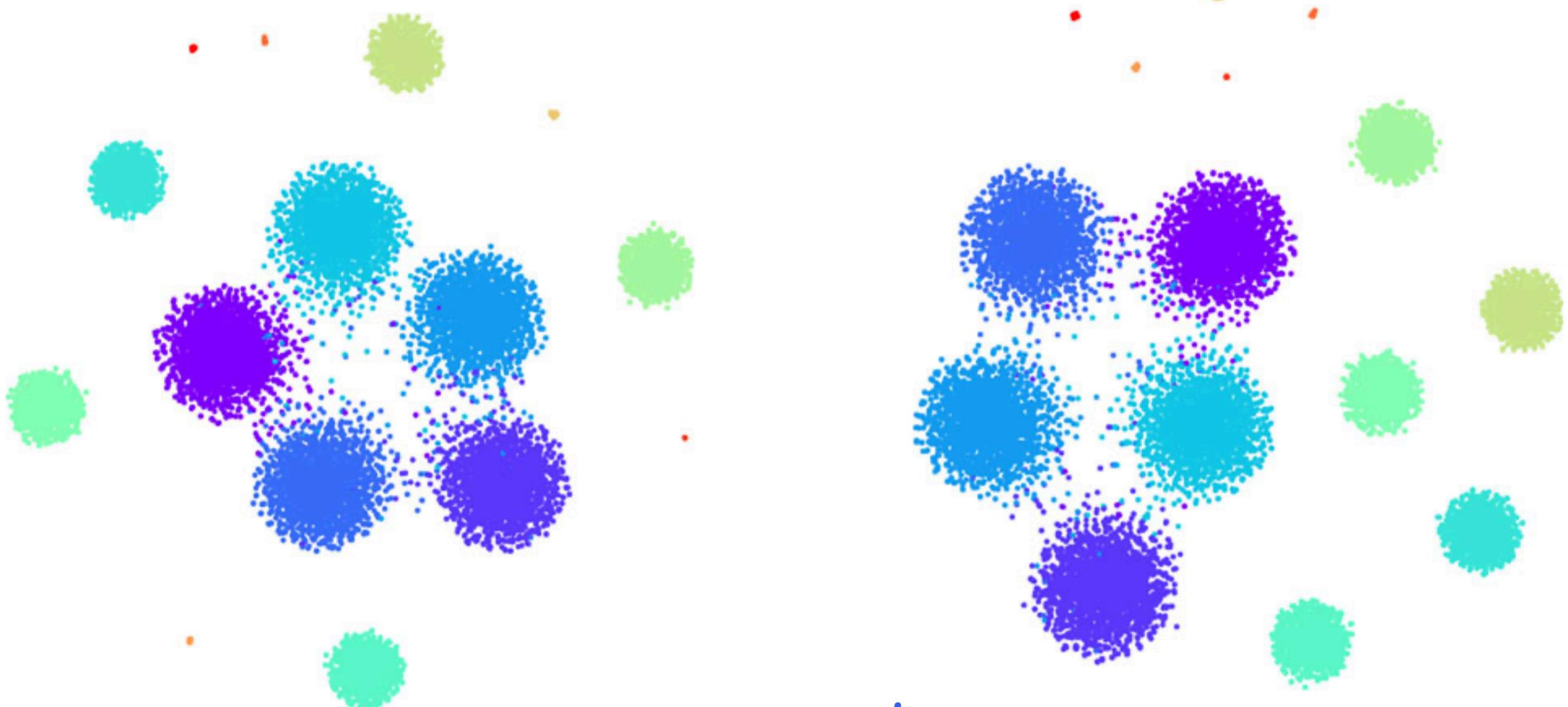
© D. Kobak, 2021

## Initialisierung



# t-SNE

## Initialisierung



Ergebnis nach zufällige  
Initialisierung

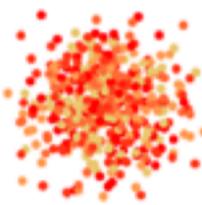
Initialisierung durch  
PCA

© Kobak & Berens, 2020

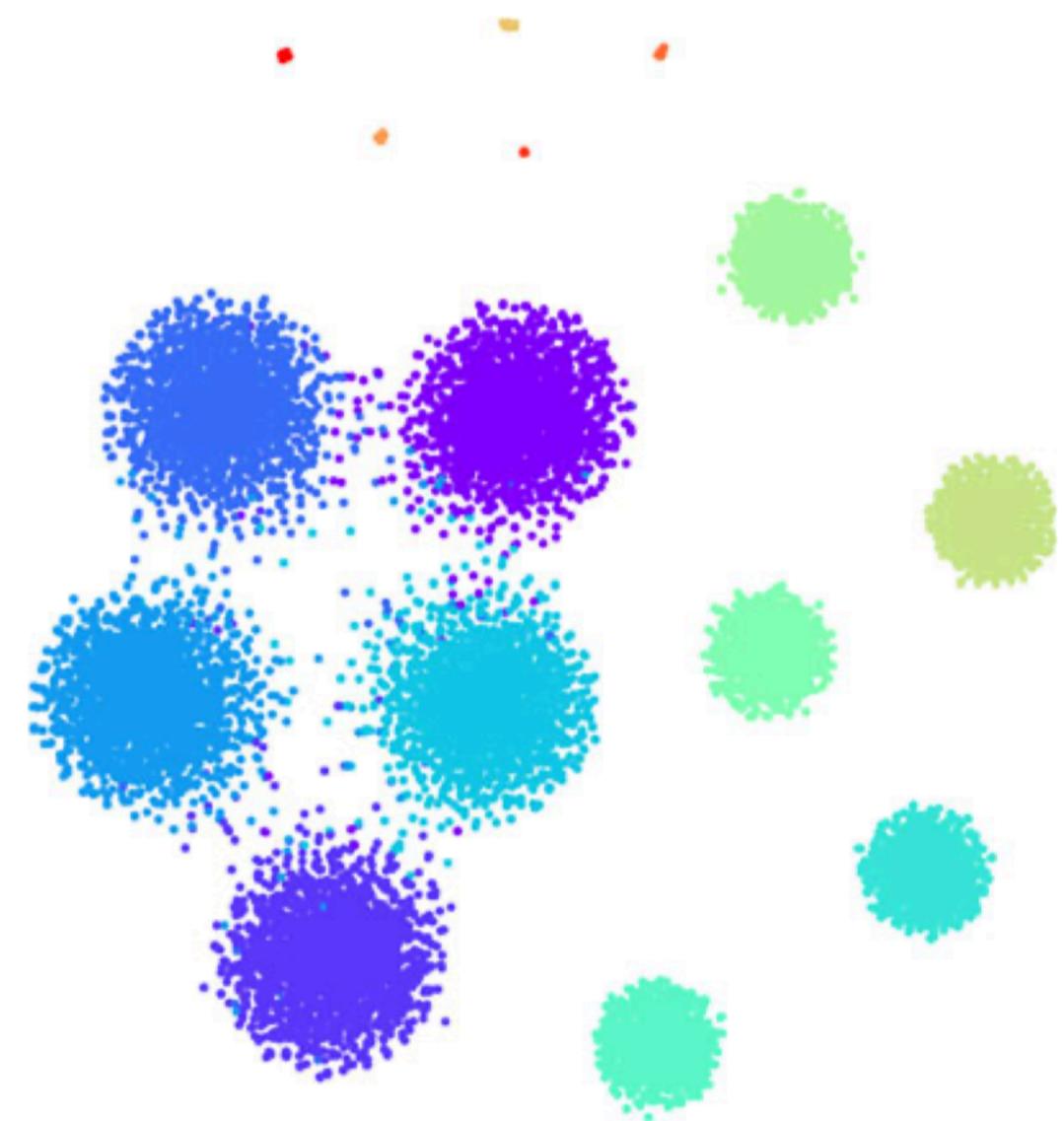
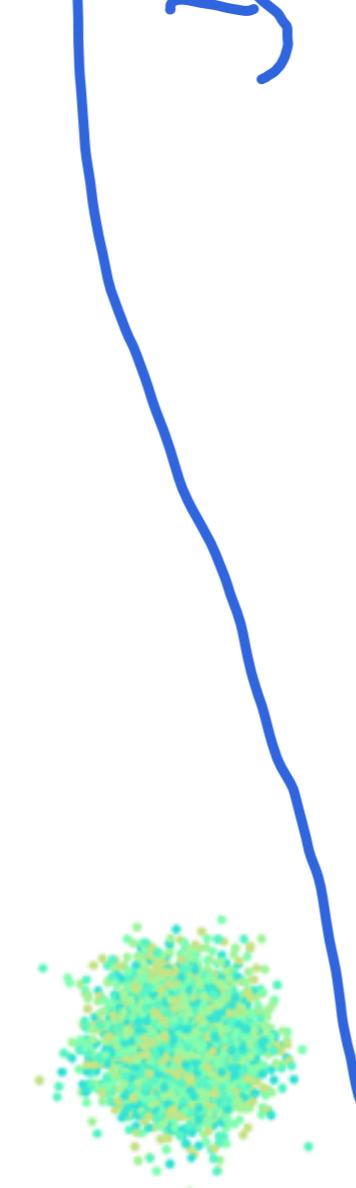
# t-SNE

## Initialisierung

PCA

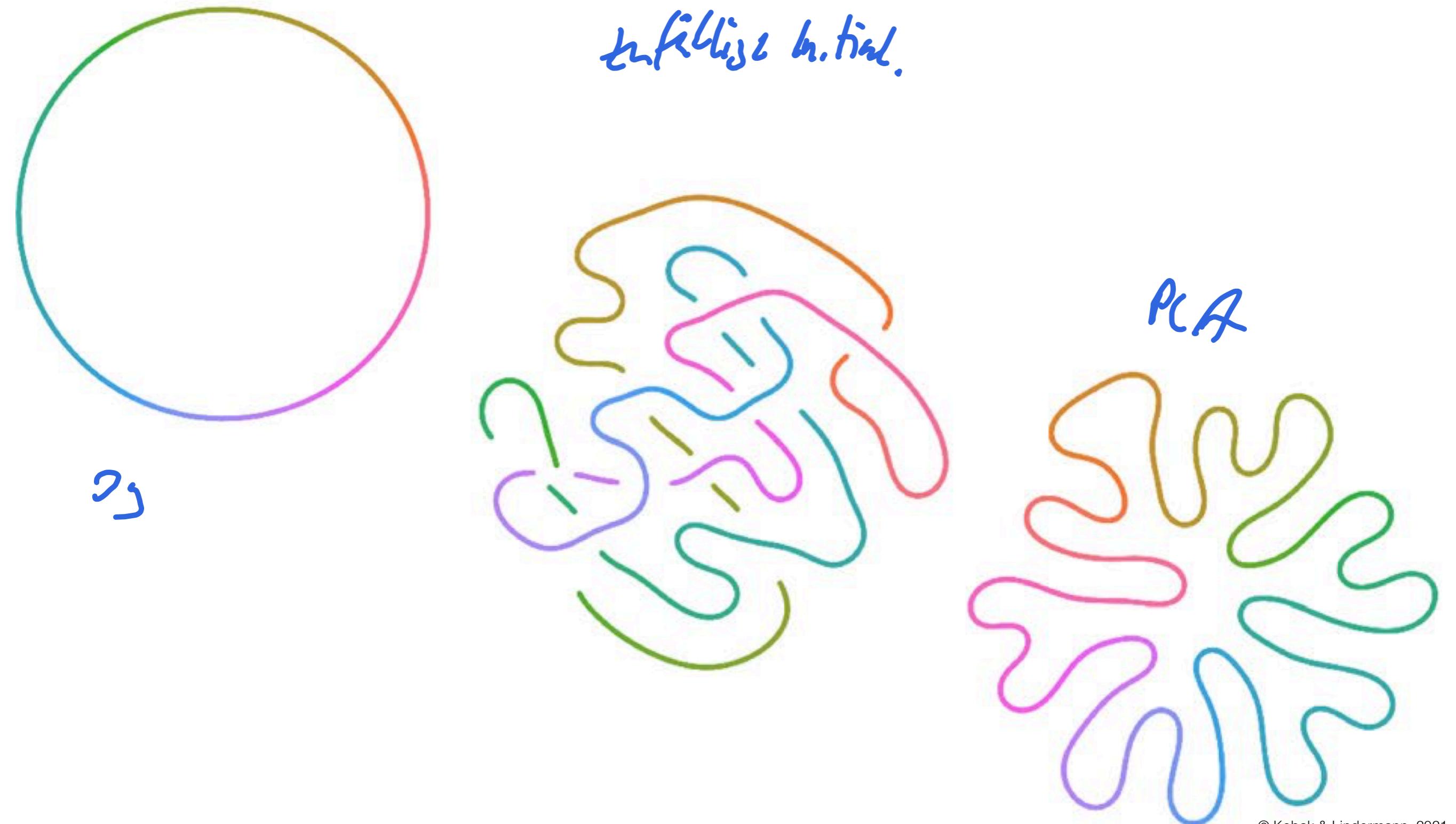


$t\text{-SNE}_2$



# t-SNE

Initialisierung



© Kobak & Lindermann, 2021

# t-SNE

## Exaggeration

$$\frac{\partial D_{\text{t-SNE}}(\rho)}{\partial z_i} \sim \sum_j p_{ij} w_{ij}(z_i - z_j) - \frac{1}{\rho Z} \sum_j w_{ij}^2(z_i - z_j)$$

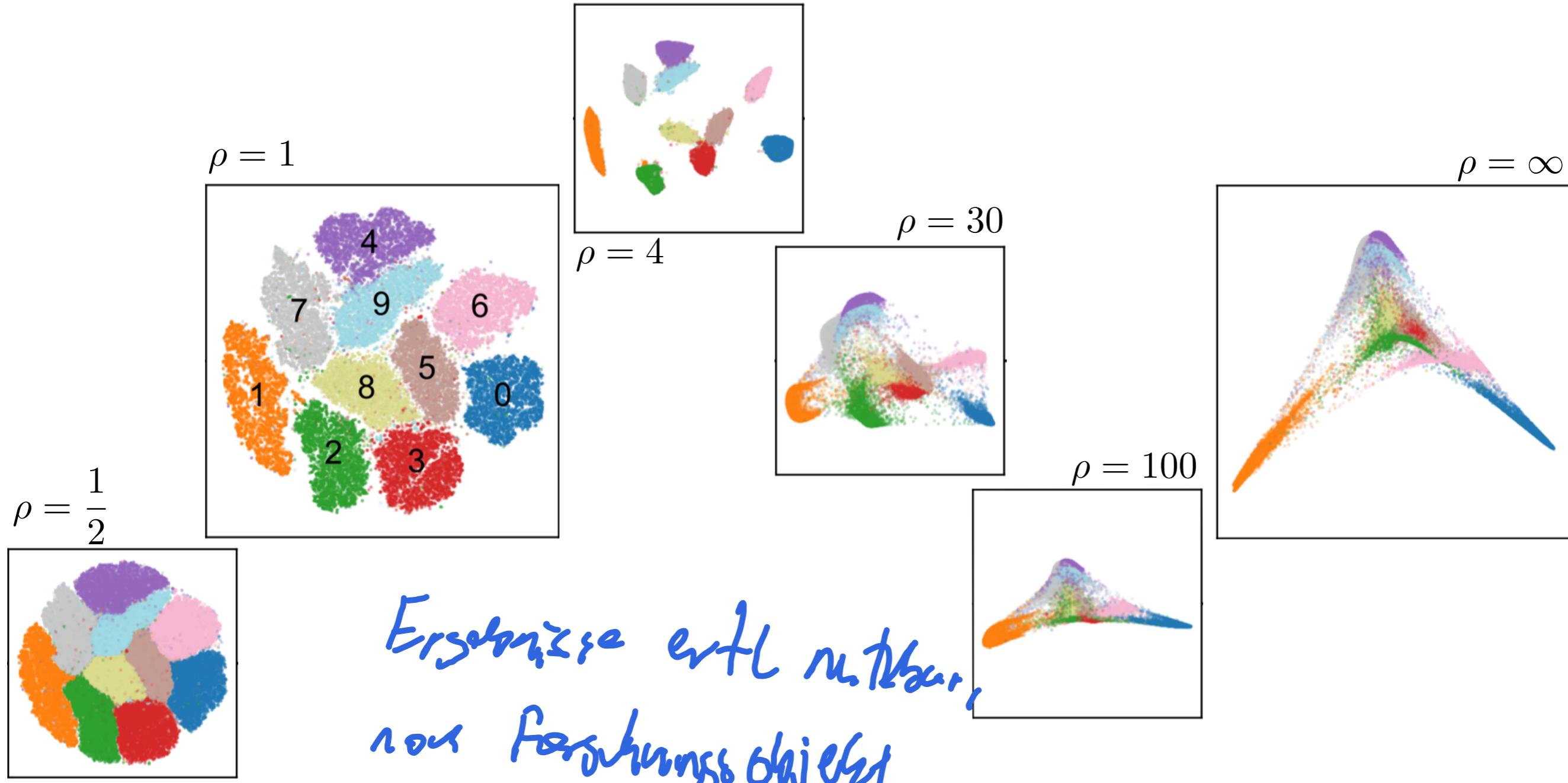
*rho*



# t-SNE

Exaggeration

$$\frac{\partial D_{\text{t-SNE}}(\rho)}{\partial z_i} \sim \sum_j p_{ij} w_{ij}(z_i - z_j) - \frac{1}{\rho Z} \sum_j w_{ij}^2(z_i - z_j)$$



- ↳ Exaggration ist wichtiger Parameter
- ↳ Perplexität auch, aber weniger als Exaggration,
- ↳ Gleichverteilung für nützliche Nachbarschaften: legitime Vereinfachung
- ↳ Initalisierung ist wichtig
  - ⇒ PCA als Preprocessing sinnvoll