universität wien

Clustering:
Balancing Abstraction and Representation

Christian Böhm, Lena G. M. Bauer, Claudia Plant

Leader of Research Group Data Mining and Machine Learning

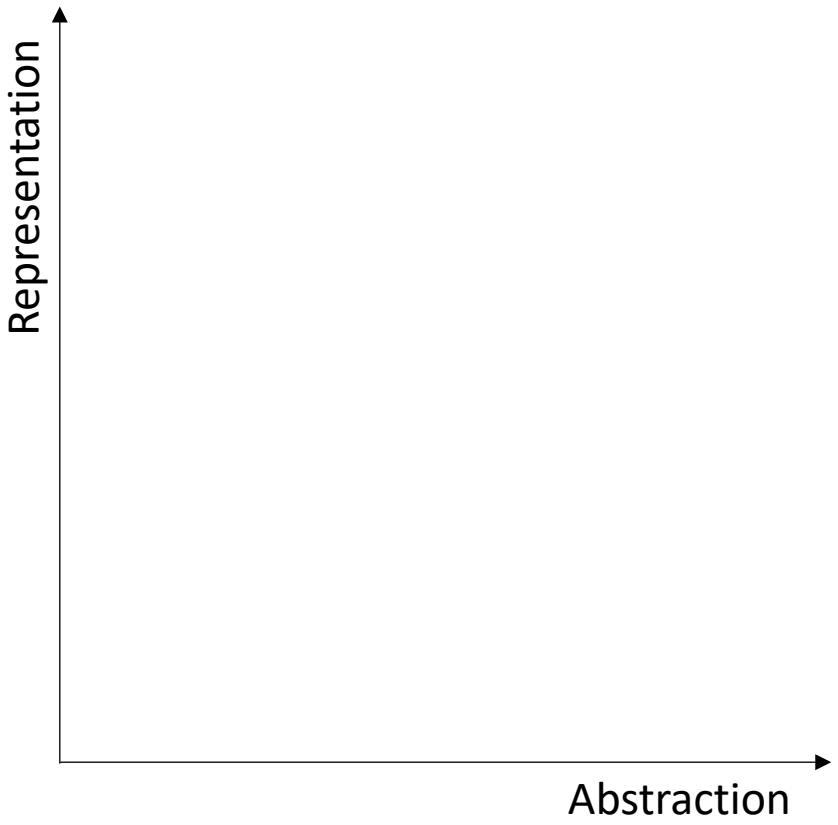
University of Vienna, Austria


70
70
70

Clustering – find a Meaningful Grouping

Alternative Clustering

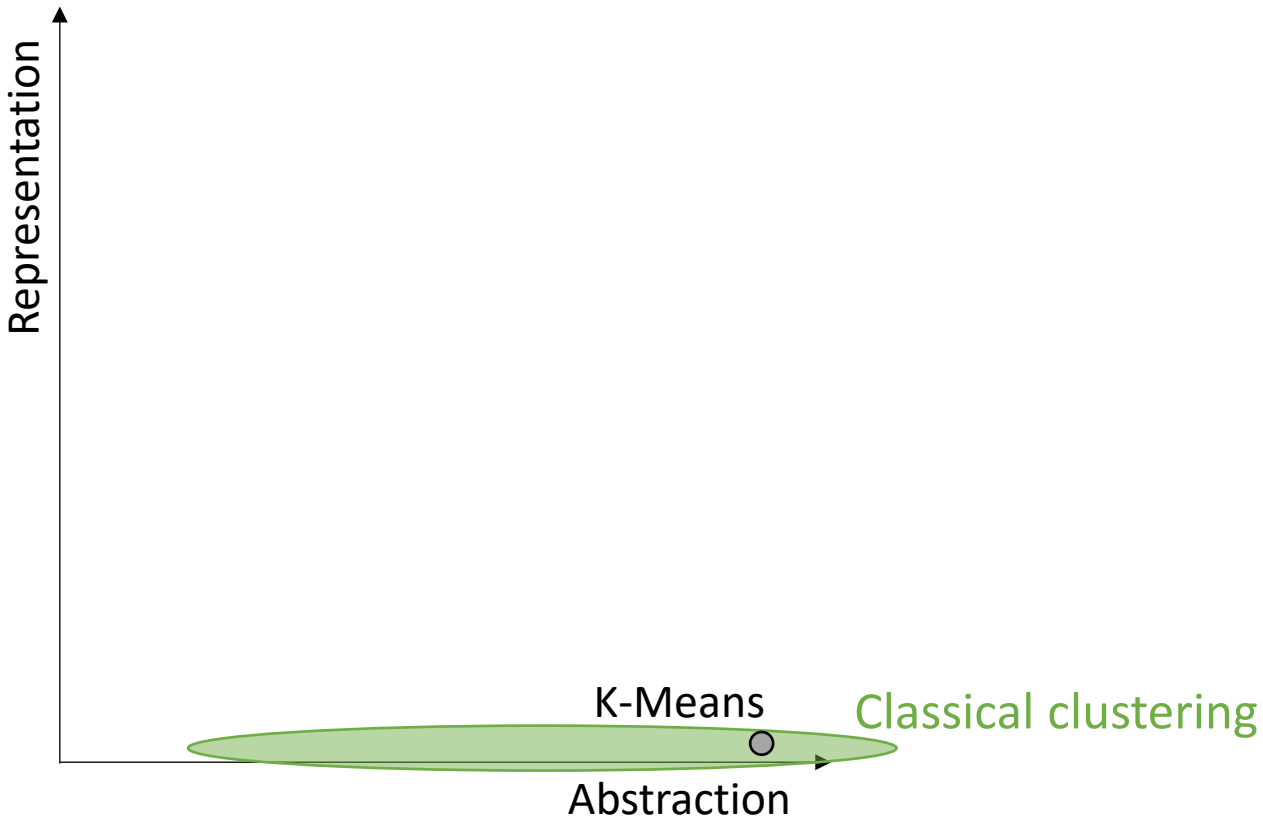
Balancing Abstraction and Representation



High representational complexity:
+ captures details
+ high accuracy
– difficult to interpret
+ little information loss

High level of abstraction:
+ generalizes
+ removes noise
+ easy to interpret
– high information loss

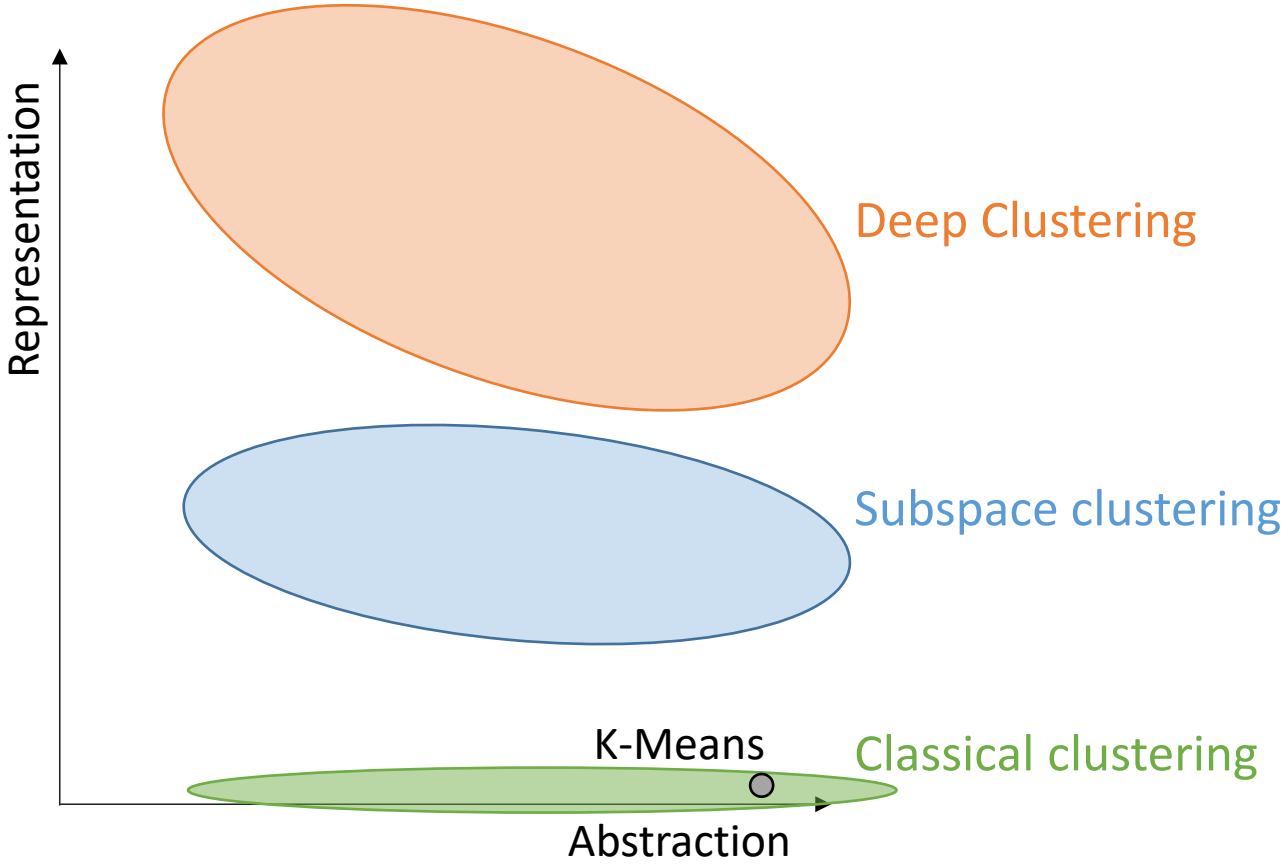
Balancing Abstraction and Representation



High representational complexity:
+ captures details
+ high accuracy
– difficult to interpret
+ little information loss

High level of abstraction:
+ generalizes
+ removes noise
+ easy to interpret
– high information loss

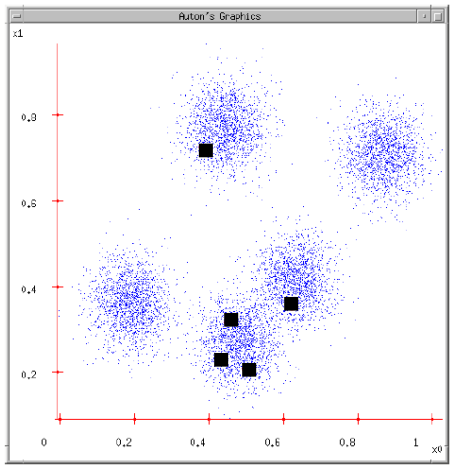
Balancing Abstraction and Representation



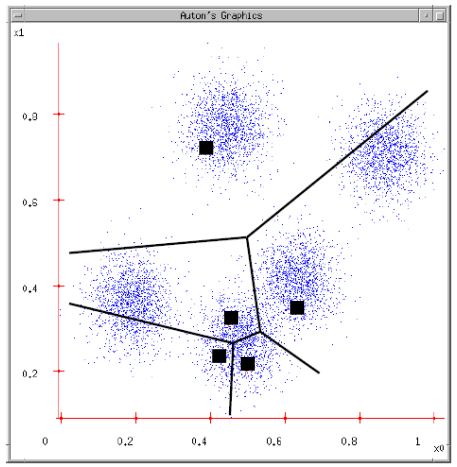
High representational complexity:
+ captures details
+ high accuracy
– difficult to interpret
+ little information loss

High level of abstraction:
+ generalizes
+ removes noise
+ easy to interpret
– high information loss

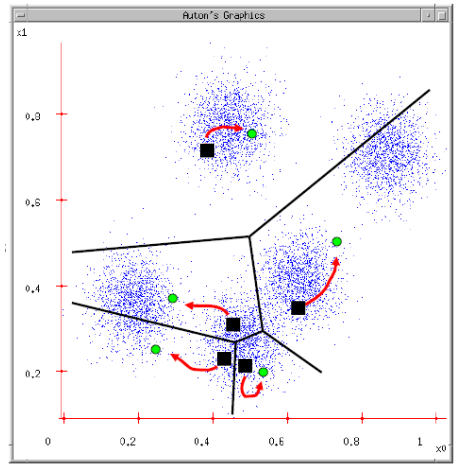
Introduction: K-Means



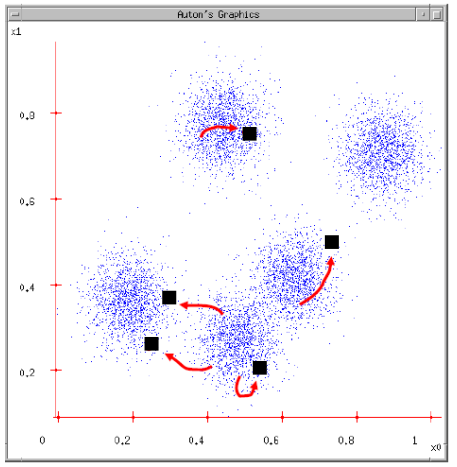
1) Random **initialization** of K cluster centers



2) **assignment** of the objects to the closest center



3) **update** of the centers



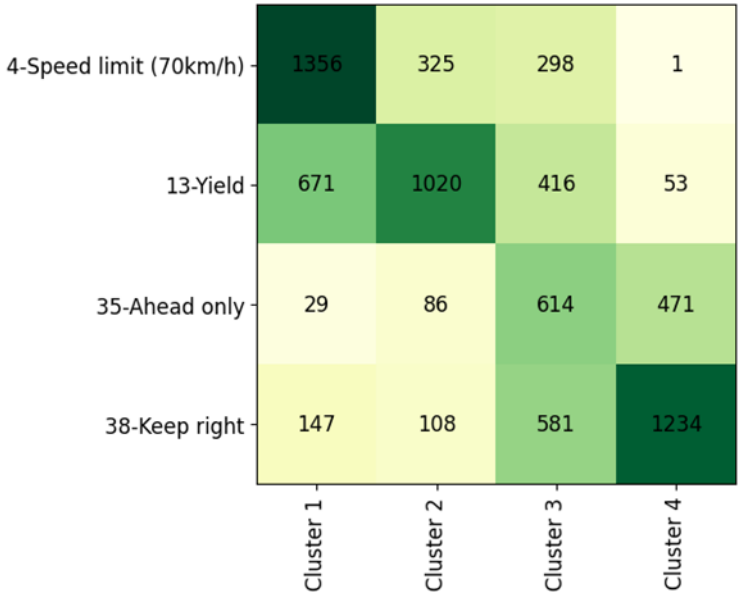
4) **iteration** of 2) and 3) until convergence

$$\mathcal{J} = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \left|\left| \mathbf{x} - \mu_i \right|\right|^2$$

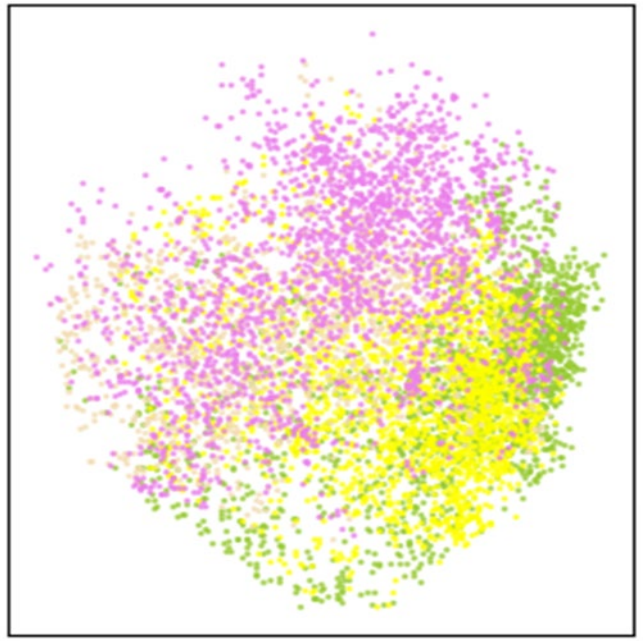
K-Means

Run this as Jupyter Notebook

NMI 0.28



Cluster centers.



PCA of the original space (3072 D) to 2D, ground truth labels.

Problems to open the notebook?

Run this as Jupyter Notebook

Find a **quick start guide** at the tutorial website https://dm.cs.univie.ac.at/research/aaai26/

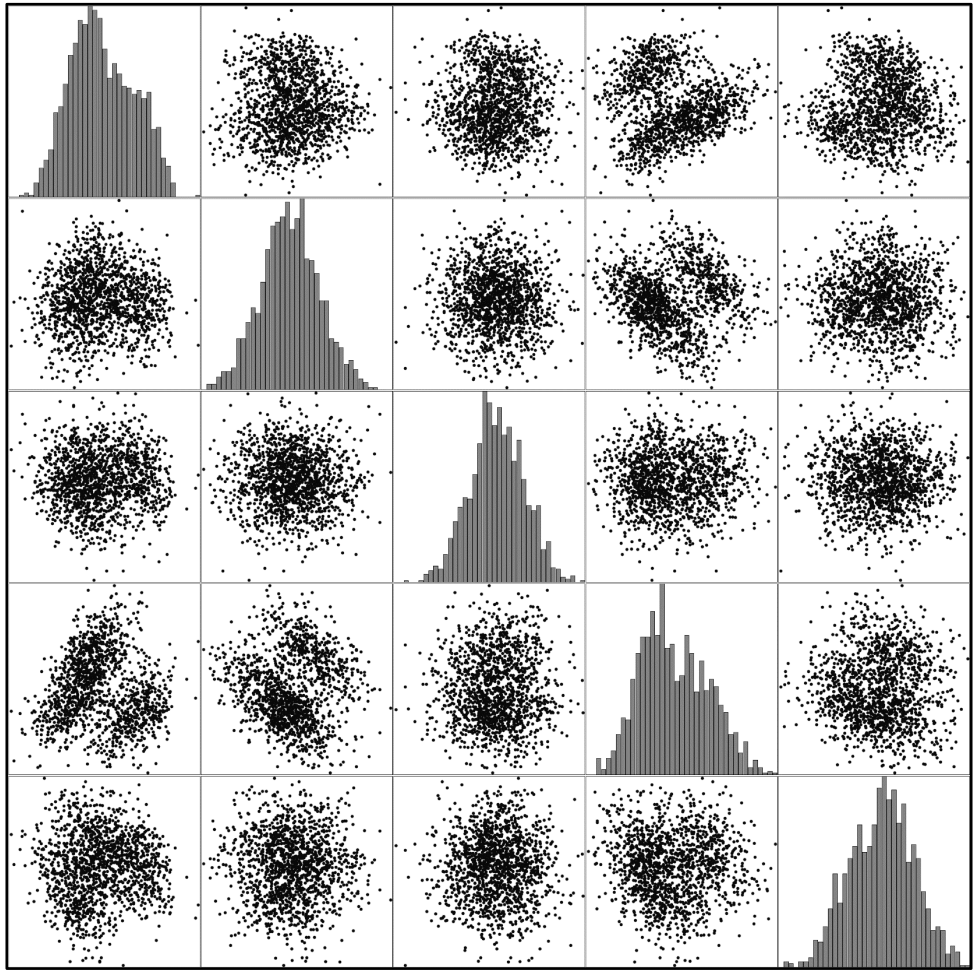
You also find there:

- The slides,
- A survey paper on the content of the tutorial including all references.

You might also visit the website later.

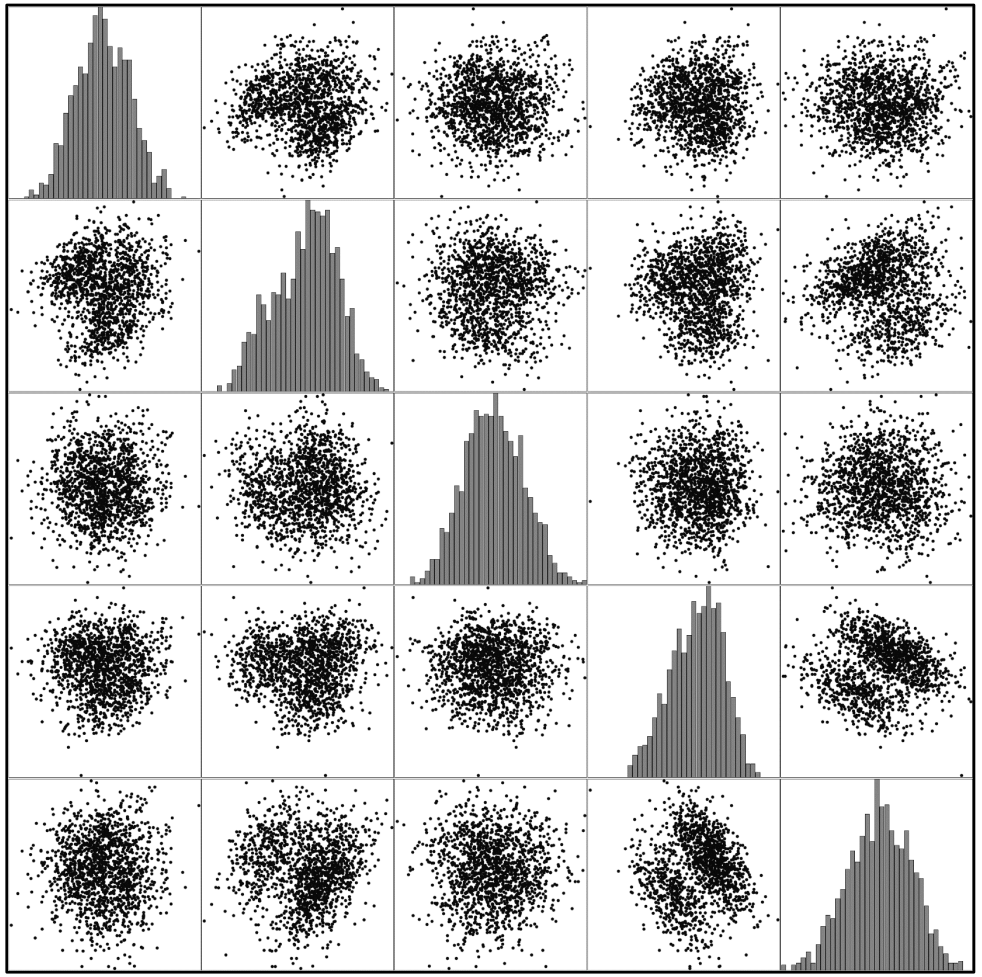
Finding the optimal subspace for K-Means

Consider this 5 D data set:



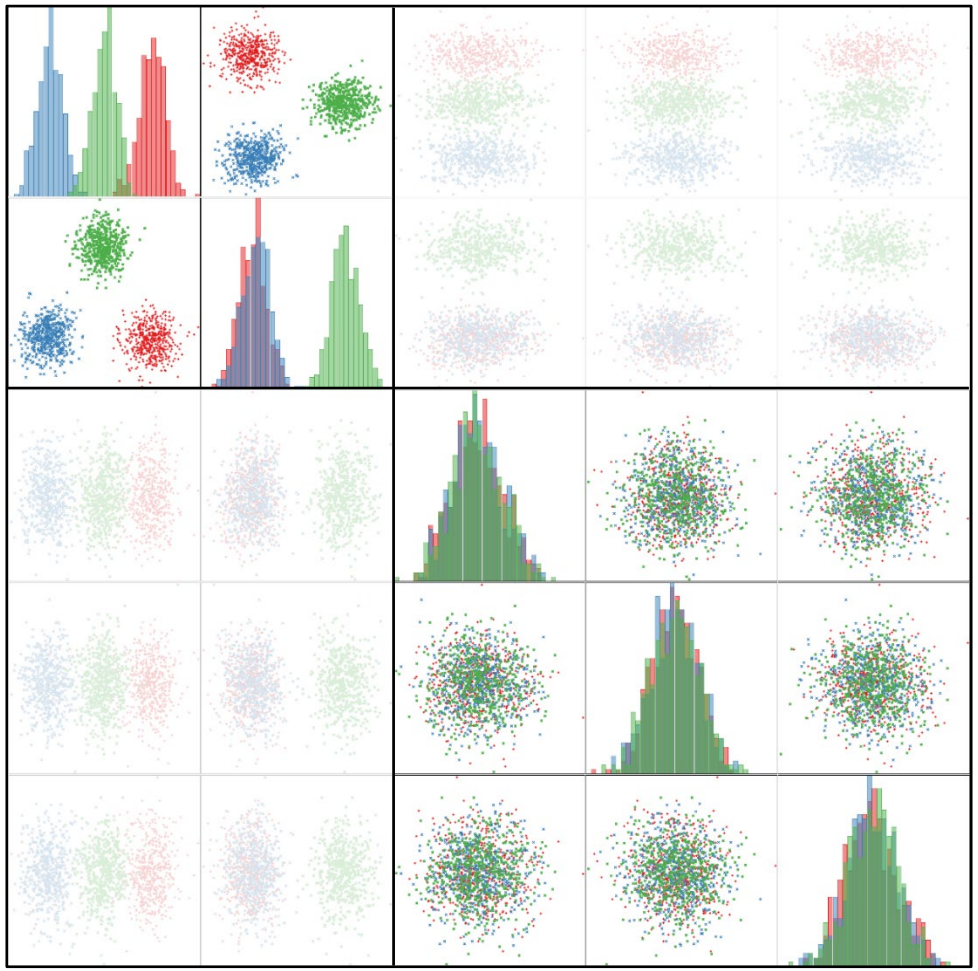
Not much cluster structure.

Let us try a PCA…

PCA:  Not better – even a bit worse.

We can do much better: Sub-K-Means

3 clear clusters in
2D „*clustered space*“



Remaining variance
in 3D „*noise space*“

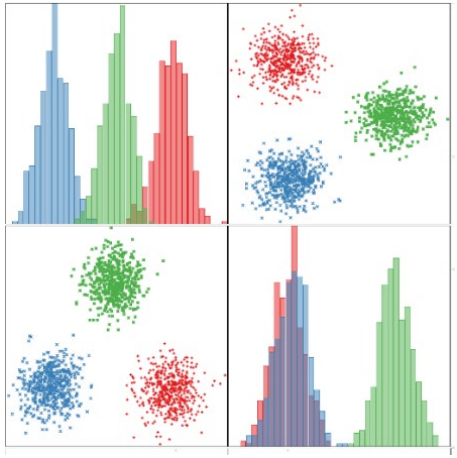
Mautz et al. Towards an Optimal Subspace for K-Means. KDD 2017.

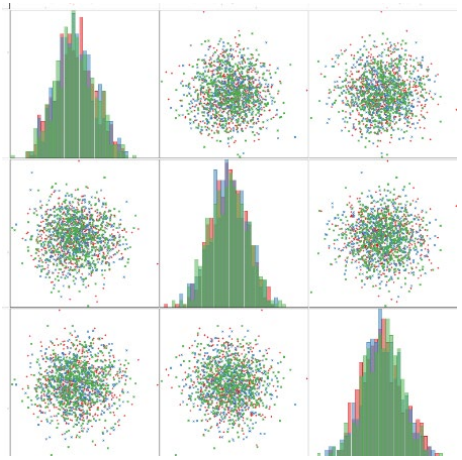
Sub-K-Means: Objective Function

$$\mathcal{J} = \left[\sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \left\| P_{\mathrm{C}}^{\mathsf{T}} V^{\mathsf{T}} \mathbf{x} - P_{\mathrm{C}}^{\mathsf{T}} V^{\mathsf{T}} \mu_i \right\|^2 \right] + \left[\sum_{\mathbf{x} \in \mathcal{D}} \left\| P_{\mathrm{N}}^{\mathsf{T}} V^{\mathsf{T}} \mathbf{x} - P_{\mathrm{N}}^{\mathsf{T}} V^{\mathsf{T}} \mu_{\mathcal{D}} \right\|^2 \right]$$

Sub-K-Means: Objective Function

$$\mathcal{J} = \underbrace{\left[\sum_{i=1}^{k}\sum_{\mathbf{x}\in C_i}\left\| P_{\mathrm{C}}^{\mathsf{T}}V^{\mathsf{T}}\mathbf{x} - P_{\mathrm{C}}^{\mathsf{T}}V^{\mathsf{T}}\mu_i \right\|^2\right]}_{\text{Clustered Space}} + \underbrace{\left[\sum_{\mathbf{x}\in\mathcal{D}}\left\| P_{\mathrm{N}}^{\mathsf{T}}V^{\mathsf{T}}\mathbf{x} - P_{\mathrm{N}}^{\mathsf{T}}V^{\mathsf{T}}\mu_{\mathcal{D}} \right\|^2\right]}_{\text{Noise Space}}$$





Algorithm Sub-K-Means

$$\mathcal{J} = \underbrace{\left[\sum_{i=1}^{k}\sum_{\mathbf{x}\in C_i}\left\| P_{\mathrm{C}}^{\mathsf{T}}V^{\mathsf{T}}\mathbf{x} - P_{\mathrm{C}}^{\mathsf{T}}V^{\mathsf{T}}\mu_i \right\|^2\right]}_{\text{Clustered Space}} + \underbrace{\left[\sum_{\mathbf{x}\in\mathcal{D}}\left\| P_{\mathrm{N}}^{\mathsf{T}}V^{\mathsf{T}}\mathbf{x} - P_{\mathrm{N}}^{\mathsf{T}}V^{\mathsf{T}}\mu_{\mathcal{D}} \right\|^2\right]}_{\text{Noise Space}}$$

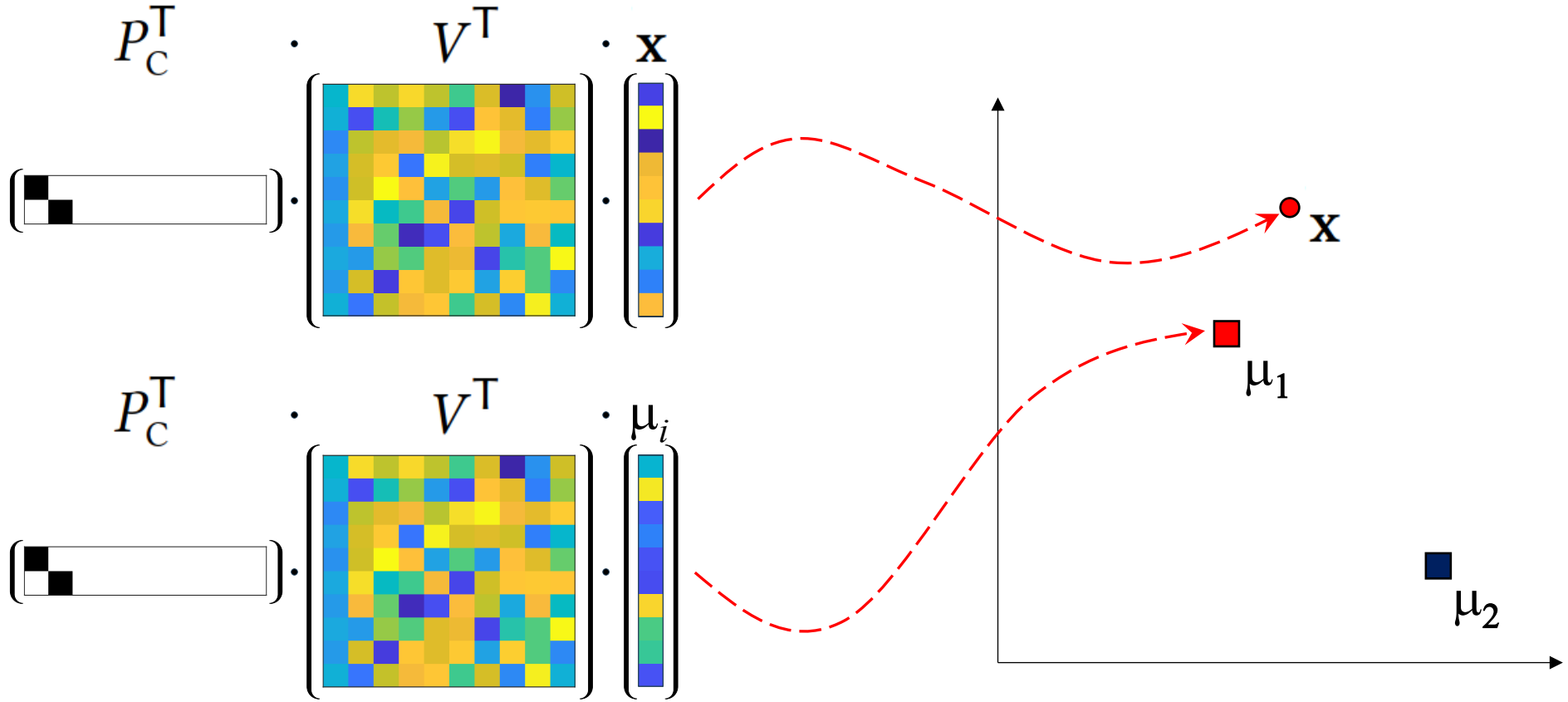
Given the number of clusters K as an input parameter

1) **Initialization** of cluster centers, rotation matrix and projection matrices
2) **Assignment** of data points to cluster centers
3) **Update** of the cluster centers, the rotation matrix and the projection matrices
4) **Iterate** 2) and 3) until **convergence**

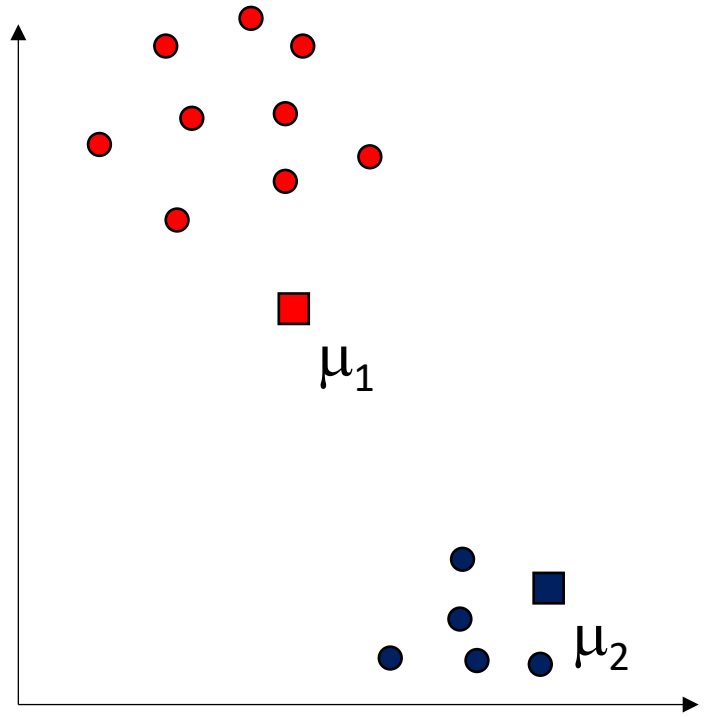
Sub-K-Means determines the dimensionality of the clustered space automatically.

Sub-K-Means: Assignment

Project all points to the clustered space and assign them to the closest center (Euclidean Distance).



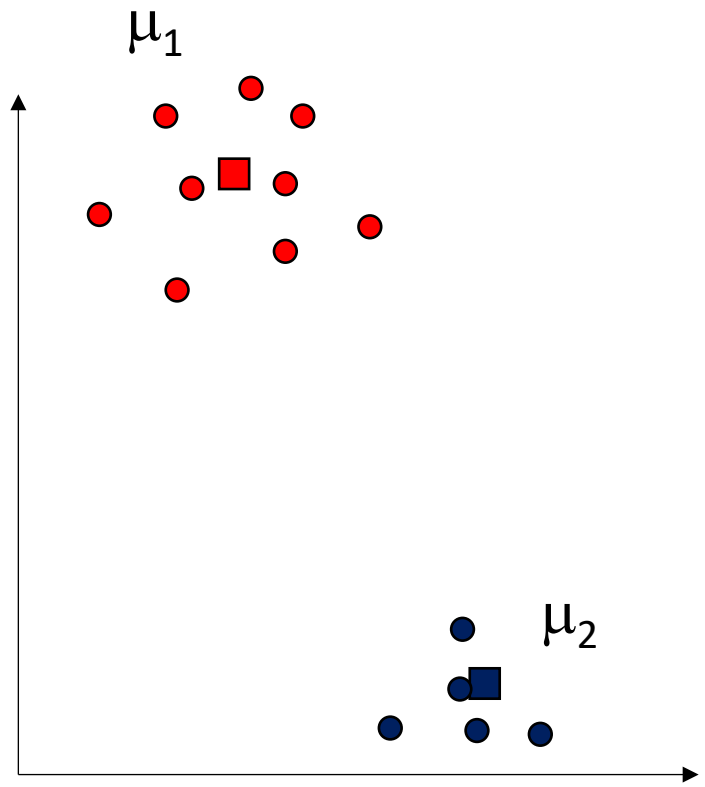
Sub-K-Means: Update of the Cluster Centers



As in classical K-Means:
The cluster center is the mean of the associated points

$$\boldsymbol{\mu}_i = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{x}.$$

Sub-K-Means: Update of the Cluster Centers



As in classical K-Means:
The cluster center is the mean of the associated points.

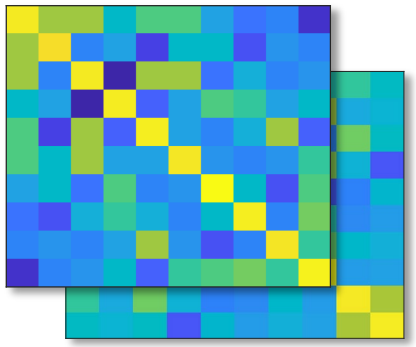
$$\boldsymbol{\mu}_i = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{x}.$$

Update of Rotation and Projection

- Is possible by solving an Eigenproblem $\operatorname{eig}\left(\left(\sum_{i=1}^{k} S_i\right) - S_{\mathcal{D}}\right)$
- Sort the Eigenvectors ascendingly according to the Eigenvalues
- Update *V*: The column vectors of V are the Eigenvectors according to this sorting
- Update P_C such that it masks the Eigenvectors corresponding to negative Eigenvalues
- Update P_N such that it masks the remaining (positive) Eigenvectors

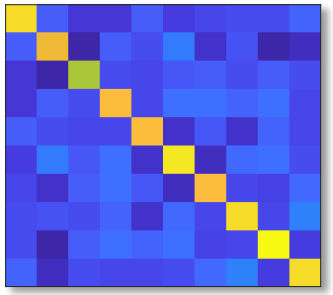
K scatter matrices of the clusters

$S_i \leftarrow \sum_{\mathbf{x} \in C_i} (\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)^{\mathsf{T}}$

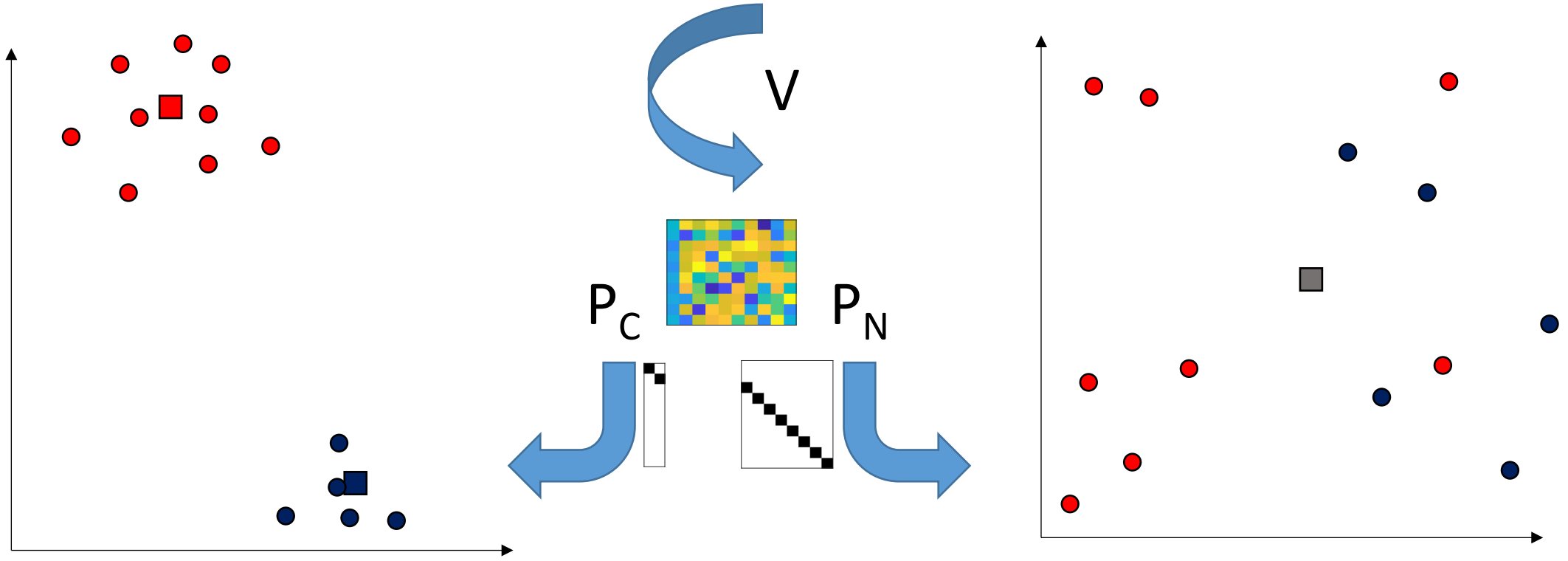


One global scatter matrix for the data

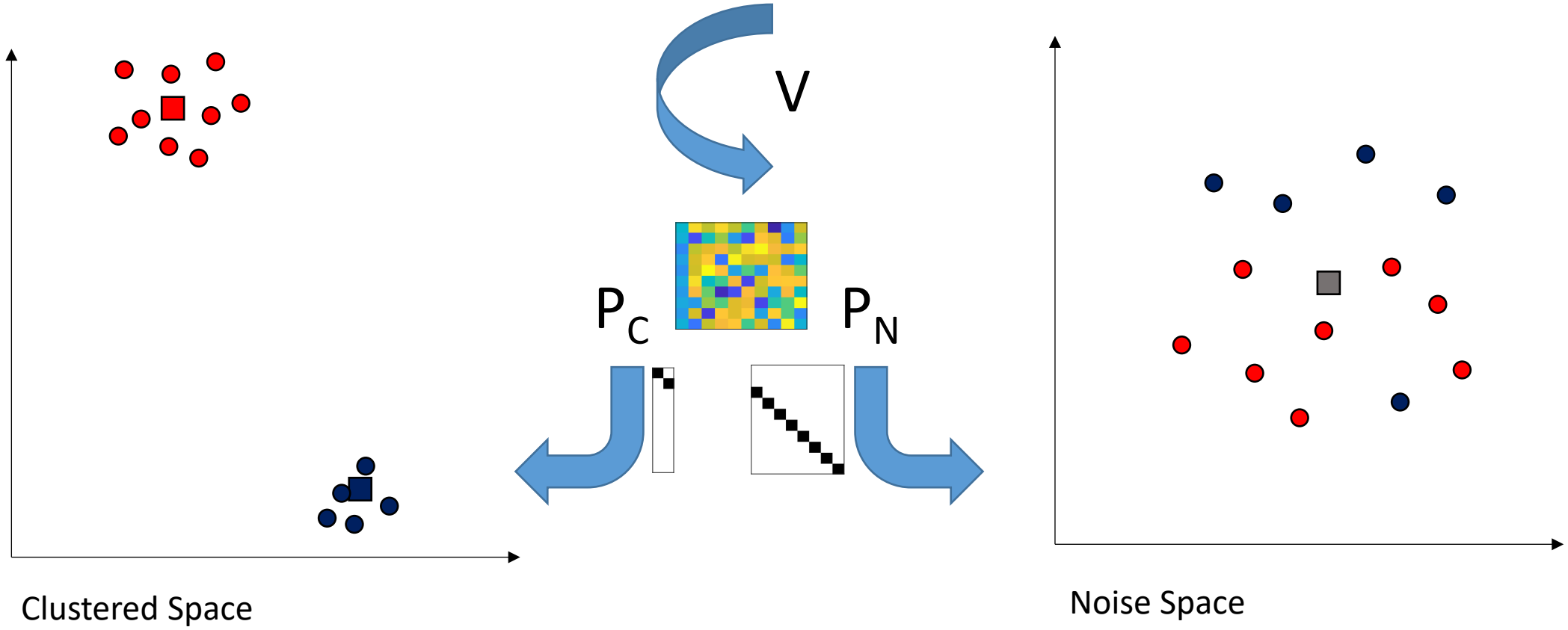
$S_{\mathcal{D}} \leftarrow \sum_{\mathbf{x} \in \mathcal{D}} (\mathbf{x} - \boldsymbol{\mu}_{\mathcal{D}})(\mathbf{x} - \boldsymbol{\mu}_{\mathcal{D}})^{\mathsf{T}}$



Intuition: Maximize multimodality in the Clustered Space; minimize it in the Noise Space



Intuition: Maximize multimodality in the Clustered Space; minimize it in the Noise Space



Proof - Sketch

Uses the Trace-Trick:

We can re-write our cost function as a trace minimization problem to obtain an Eigenproblem

$$\begin{aligned}\mathcal{J} &= \left[\sum_{i=1}^{k}\sum_{\mathbf{x}\in C_i}\left|\left|P_{\mathrm{C}}^{\mathsf{T}}V^{\mathsf{T}}\mathbf{x} - P_{\mathrm{C}}^{\mathsf{T}}V^{\mathsf{T}}\mu_i\right|\right|^2\right] + \left[\sum_{\mathbf{x}\in\mathcal{D}}\left|\left|P_{\mathrm{N}}^{\mathsf{T}}V^{\mathsf{T}}\mathbf{x} - P_{\mathrm{N}}^{\mathsf{T}}V^{\mathsf{T}}\mu_{\mathcal{D}}\right|\right|^2\right] \\ &= \mathrm{Tr}\left(P_{\mathrm{C}}\,P_{\mathrm{C}}^{\mathsf{T}}V^{\mathsf{T}}\left(\left(\sum_{i=1}^{k}S_i\right) - S_{\mathcal{D}}\right)V\right) + \mathrm{Tr}\left(V^{\mathsf{T}}S_{\mathcal{D}}V\right)\end{aligned}$$

Proof - Sketch

Uses the Trace-Trick:

We can re-write our cost function as a trace minimization problem to obtain an Eigenproblem

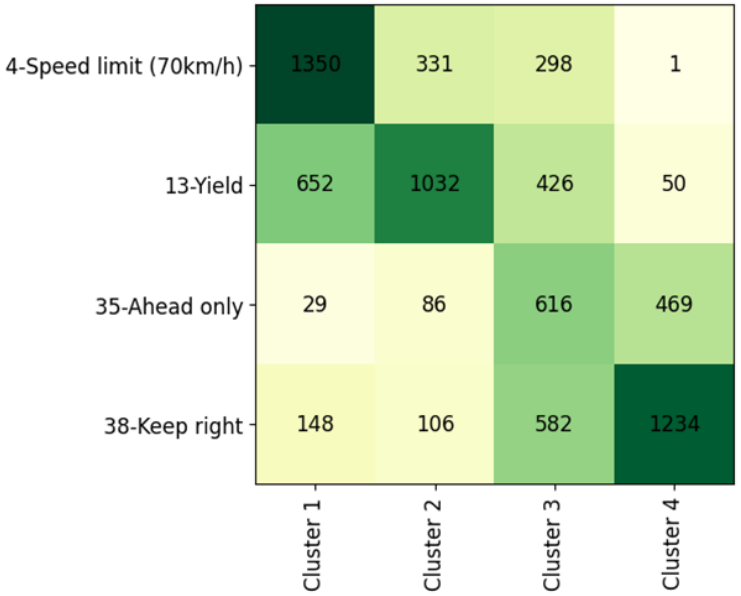
$$\begin{aligned}\mathcal{J} &= \left[\sum_{i=1}^{k}\sum_{\mathbf{x}\in C_i}\left|\left|P_{\mathrm{C}}^{\mathsf{T}}V^{\mathsf{T}}\mathbf{x} - P_{\mathrm{C}}^{\mathsf{T}}V^{\mathsf{T}}\mu_i\right|\right|^2\right] + \left[\sum_{\mathbf{x}\in\mathcal{D}}\left|\left|P_{\mathrm{N}}^{\mathsf{T}}V^{\mathsf{T}}\mathbf{x} - P_{\mathrm{N}}^{\mathsf{T}}V^{\mathsf{T}}\mu_{\mathcal{D}}\right|\right|^2\right] \\ &= \mathrm{Tr}\left(P_{\mathrm{C}}\,P_{\mathrm{C}}^{\mathsf{T}}V^{\mathsf{T}}\left(\left(\sum_{i=1}^{k}S_i\right) - S_{\mathcal{D}}\right)V\right) + \mathrm{Tr}\left(V^{\mathsf{T}}S_{\mathcal{D}}V\right)\end{aligned}$$

Automatic selection of dimensionality:

We minimize our objective function by assigning Eigenvectors with negative Eigenvalues to the Clustered Space and the rest to the Noise Space.

Sub-K-Means

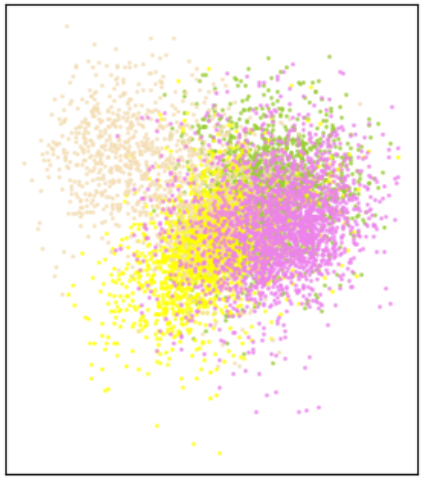
NMI 0.28



Cluster centers.

Cluster space first 2 dimensions

Noise space first 2 dimensions

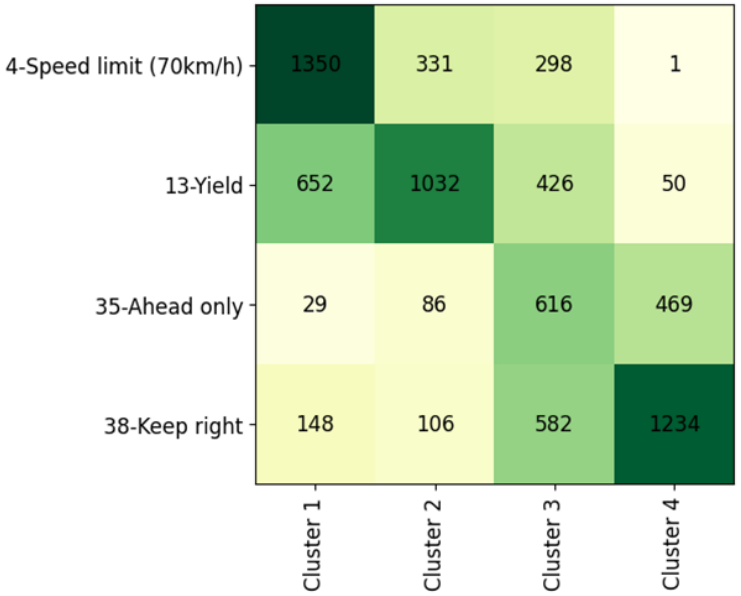


4-Speed limit (70km/h) 13-Yield 35-Ahead only 38-Keep right

Ground truth labels.

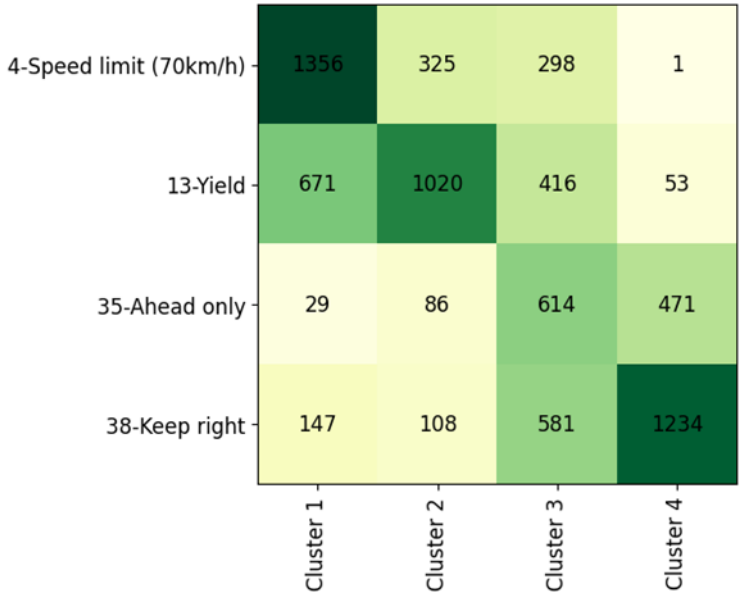
Sub-K-Means vs K-Means: Very Similar Results

Sub-K-Means: NMI 0.28



Cluster centers.

K-Means: NMI 0.28



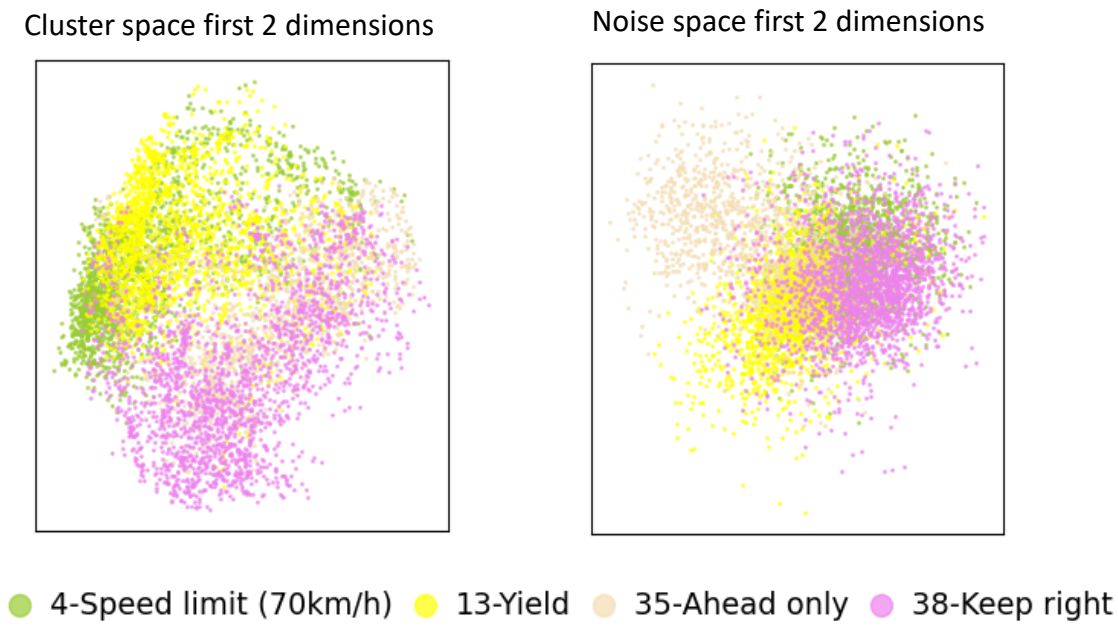
Cluster centers.

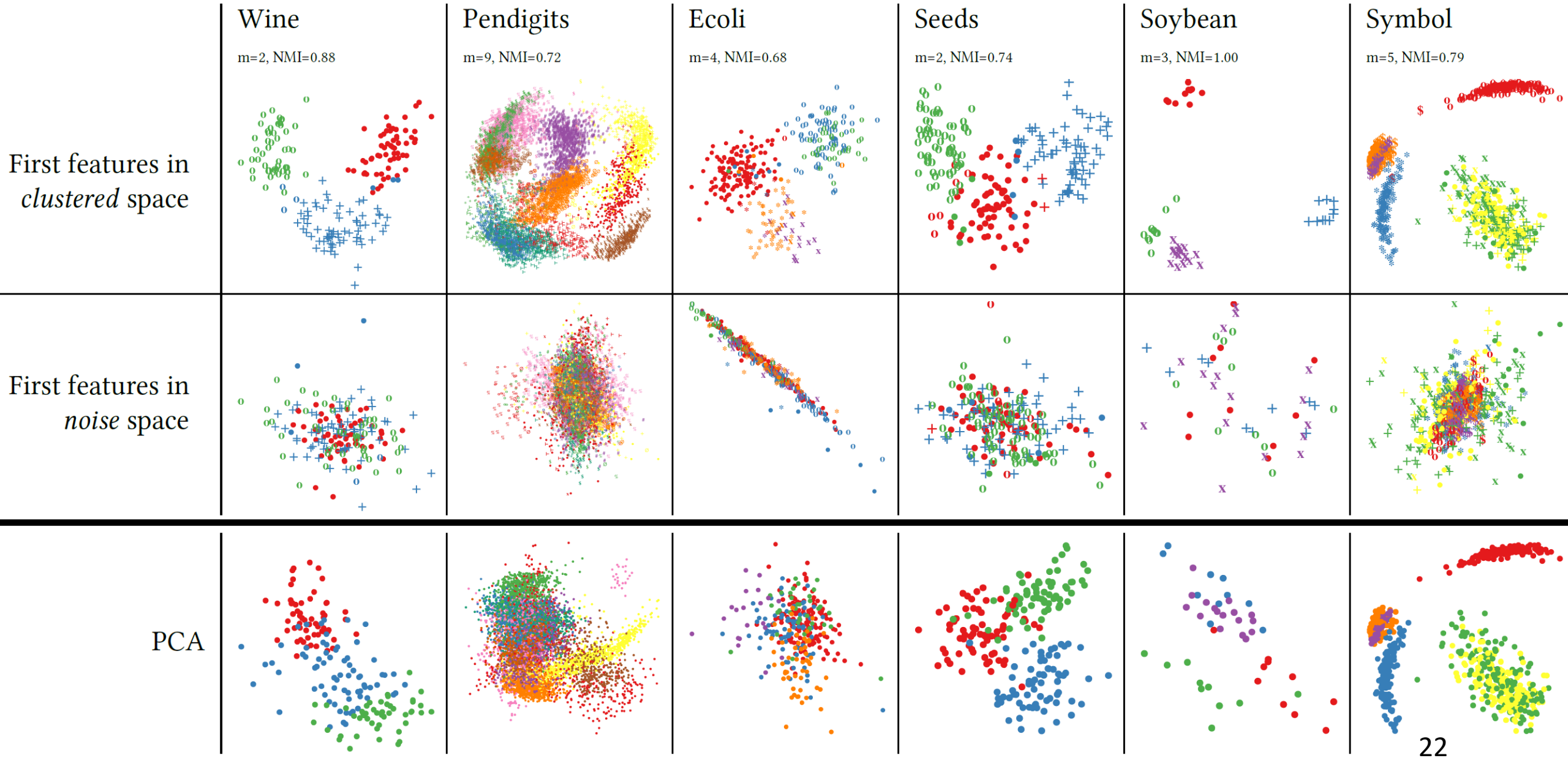
Clustering Needs a Suitable Representation

High dimensionality of the clustered space.
Clustered space: 1450 dimensions;
Noise space: 1622 dimensions.

Sub-K-Means works best when the clustered space is low-dimensional.

Linear space transformation is most effective on data with moderate dimensionality.




Wine
m=2, NMI=0.88
Pendigits
m=9, NMI=0.72
Ecoli
m=4, NMI=0.68
Seeds
m=2, NMI=0.74
Soybean
m=3, NMI=1.00
Symbol
m=5, NMI=0.79
First features in *clustered* space
First features in *noise* space
PCA

Summary: Subspace Clustering

Representation: Subspaces either axis-parallel or arbitrarily oriented

Common subspace clustering: one subspace for the clustering, e.g., Sub-K-Means

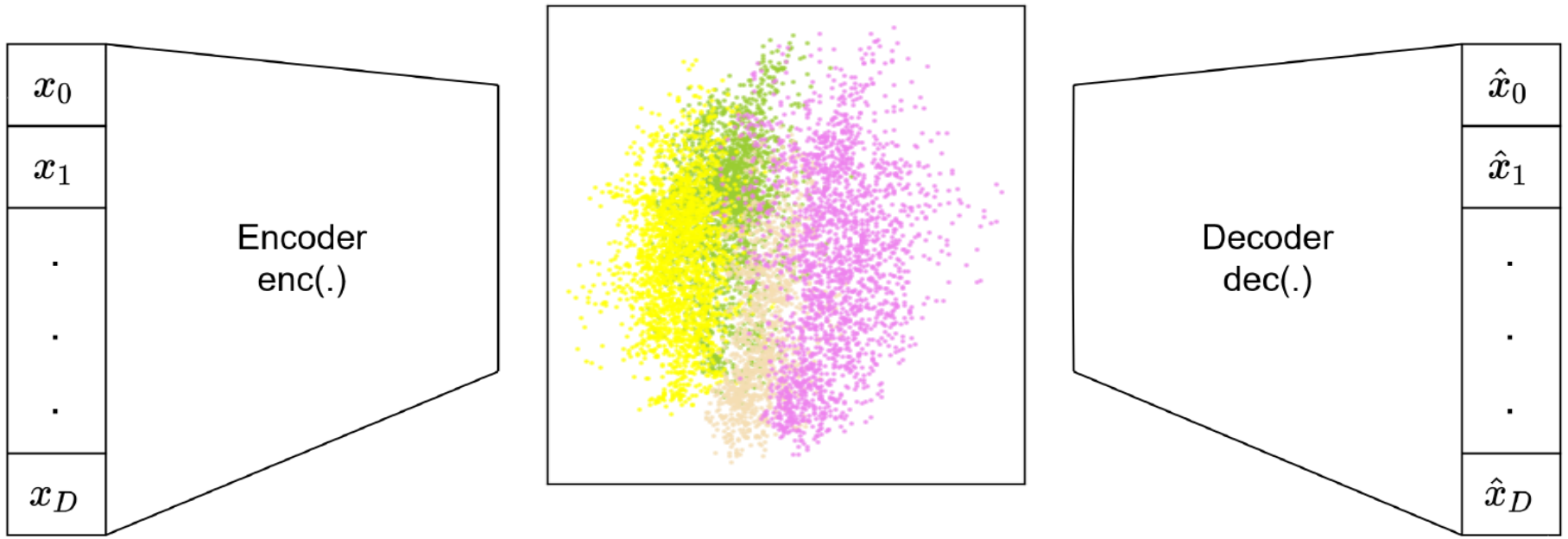
Alternative subspace clustering: multiple subspaces with different clusterings, e.g., NR-K-Means

Most other methods: Individual subspace for each cluster, e.g., 4C

Mautz et al. Towards an Optimal Subspace for K-Means. KDD 2017
Mautz et al. Discovering Non-redundant K-Means Clusterings in Optimal Subspaces. KDD 2018
Böhm et al. Computing Clusters of Correlation Connected Objects. SIGMOD 2018

Moving to Higher Dimensions by Deep Learning

Basis: deep autoencoder



$$\mathcal{L}_{\mathrm{R}} = \sum_{x \in \mathcal{D}} \left|\left| \mathbf{x} - \mathrm{dec}\big(\mathrm{enc}(\mathbf{x})\big) \right|\right|_2^2$$

Autoencoder: Original and Reconstructed Images

Original:

Reconstructed:

Sequential Deep Clustering Approach

1) Use an autoencoder to learn a non-linear embedding of your data
   i.e., Feature learning/Representation learning

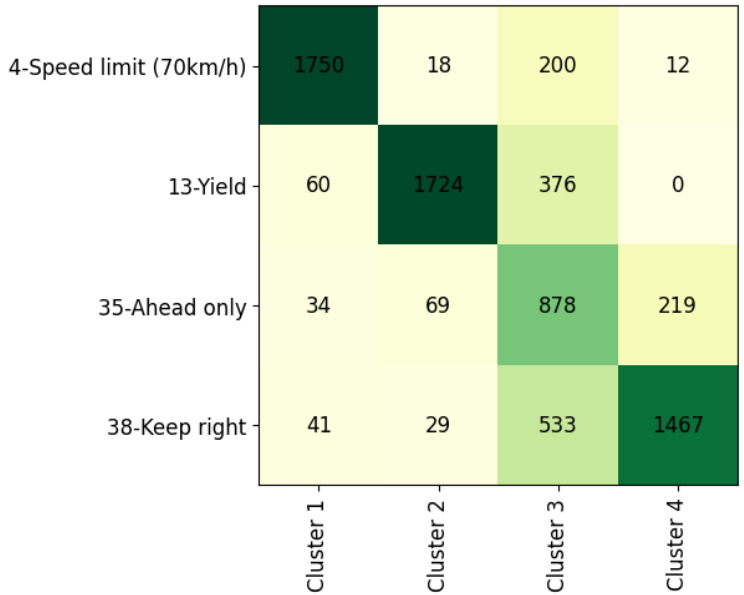
1) Cluster that data with some algorithm of your choice, e.g. K-Means

This is not a bad idea and often useful, but it might limit our solution

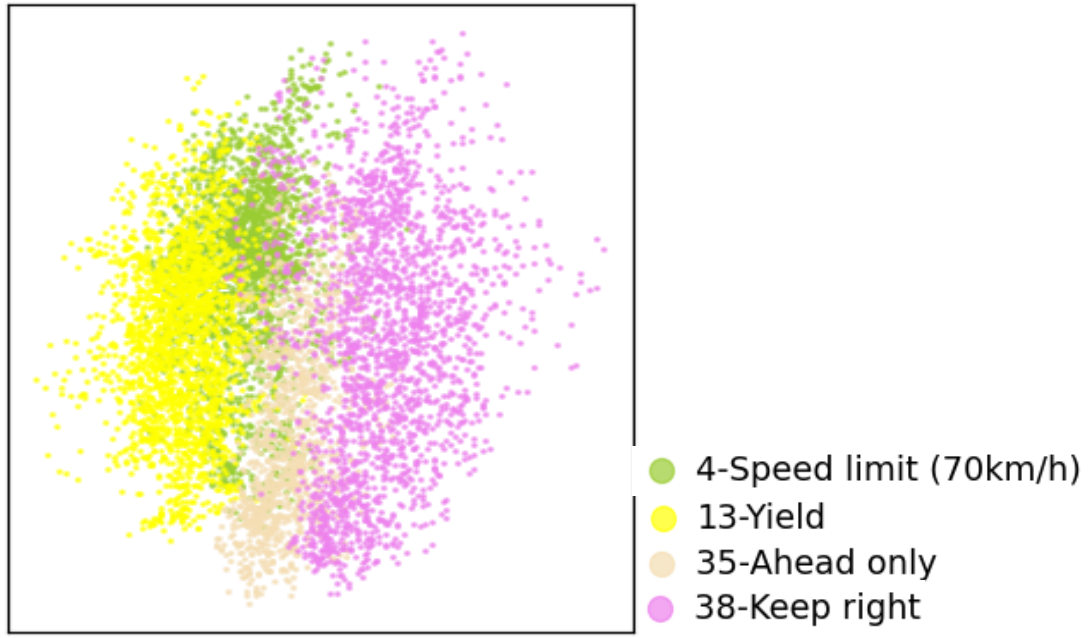
-> We are stuck to the initial representation

Autoencoder Followed by K-Means

NMI 0.56

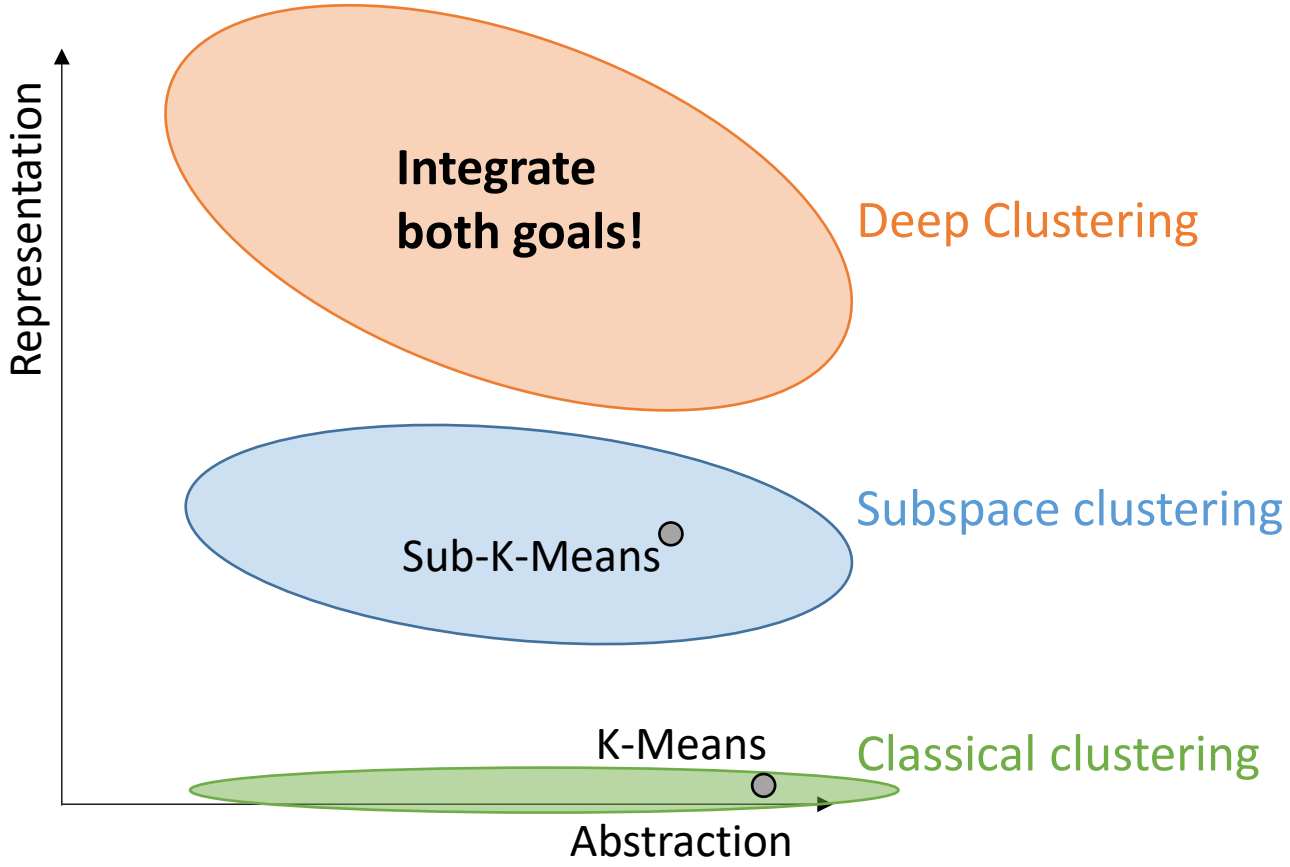


Cluster centers.



PCA of the 10D latent space to 2D.

Balancing Abstraction and Representation



High representational complexity:
+ captures details
+ high accuracy
– difficult to interpret
+ little information loss

High level of abstraction:
+ generalizes
+ removes noise
+ easy to interpret
– high information loss

Centroid-based Deep Clustering Paradigm

1) **Pre-training:** Use an autoencoder to learn a non-linear embedding of your data
2) **Obtain initial cluster labels**; typically with K-Means
3) **Deep clustering:** Refine the embedding and the clustering

DEC - Deep Embedded Clustering

Objective Function:

KL divergence between „hard“ target distribution P and the cluster assignment Matrix Q:

$$\text{Minimize} \quad KL(P||Q) = \sum_{i=1}^{n} \sum_{j=1}^{k} p_{i,j} \cdot \log\left(\frac{p_{i,j}}{q_{i,j}}\right)$$

Enforce a latent space with clear cluster assignments

Xie et al. Deep Embedding for Clustering Analysis. ICML 2016

DEC – Cluster Assignments

$Q_{N\times K}$ is the matrix of soft assignments $q_{i,j}$ of N data points to the K centroids

$$q_{i,j} = \frac{\left(1 + || \operatorname{enc}(\mathbf{x}_i) - \mu_j ||_2^2\right)^{-1}}{\sum_{j' \neq j} \left(1 + || \operatorname{enc}(\mathbf{x}_i) - \mu_{j'} ||_2^2\right)^{-1}}$$

$q_{i,j}$ are soft assignments of the i^{th} data point to the j^{th} cluster centroid

DEC – Target Distribution for Cluster Assignments

Target distribution $P_{N\times K}$

Desirable properties:

- strengthen predictions on data points assigned with high confidence
- normalize loss contribution for each centroid

$$p_{i,j} = \frac{\frac{q_{i,j}^2}{\sum_i q_{i,j}}}{\sum_{j' \neq j} \frac{q_{i,j'}^2}{\sum_i q_{i,j'}}}$$

DEC – Target Distribution for Cluster Assignments

Target distribution $P_{N\times K}$

Desirable properties:

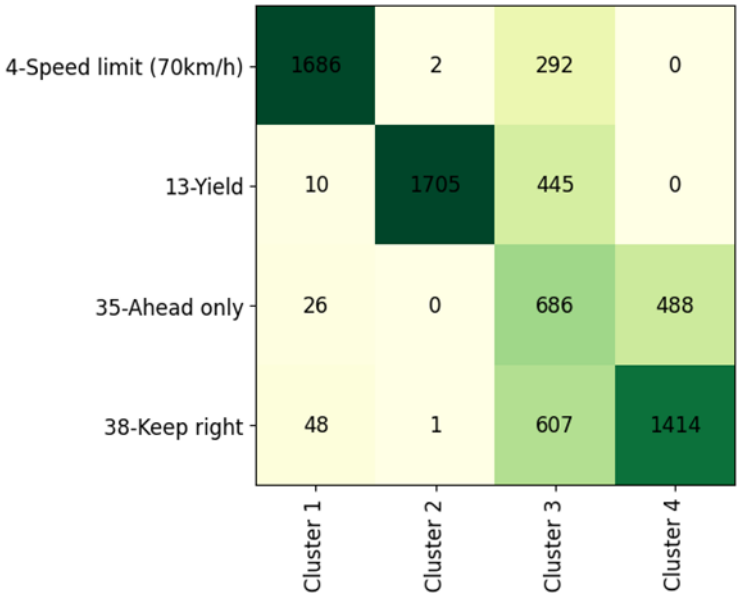
- strengthen predictions on data points assigned with high confidence
- normalize loss contribution for each centroid

By minimizing $KL(P||Q)$ the algorithm is forced to cluster the latent space, i.e. to abstract.

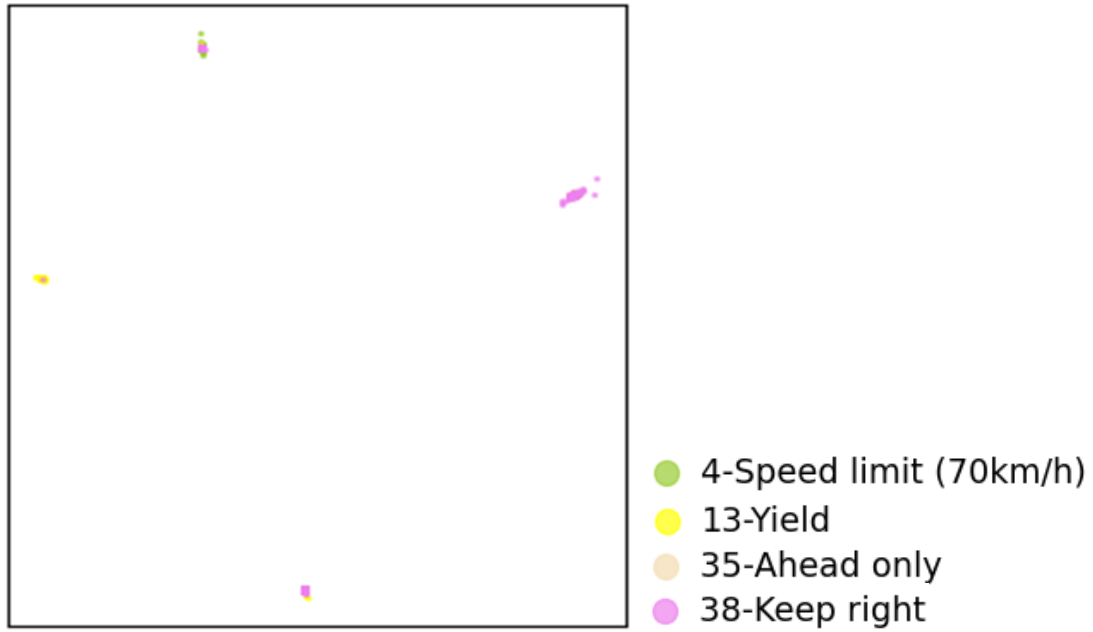
$$p_{i,j} = \frac{\frac{q_{i,j}^2}{\sum_i q_{i,j}}}{\sum_{j' \neq j} \frac{q_{i,j'}^2}{\sum_i q_{i,j'}}}$$

DEC

NMI 0.58



Cluster centers.



PCA of latent space to 2D.

DEC: Original and Reconstructed Images

Original:

Reconstructed:

IDEC-Improved Deep Embedded Clustering

It is not wise to forget the goal of representation learning during the deep clustering phase.

Loss function:

$$\mathcal{L} = \mathcal{L}_{\mathrm{R}} + \lambda \cdot \mathcal{L}_{\mathrm{C}}$$

Reconstruction Loss $$\mathcal{L}_{\mathrm{R}} = \sum_{x \in \mathcal{D}} \left|\left| \mathbf{x} - \mathrm{dec}\big(\mathrm{enc}(\mathbf{x})\big) \right|\right|_2^2$$

Compression Loss $$\mathcal{L}_{\mathrm{C}} = KL(P||Q)$$

Guo, et al: Improved Deep Embedded Clustering with Local Structure Preservation. IJCAI 2017

IDEC-Improved Deep Embedded Clustering

It is not wise to forget the goal of representation learning during the deep clustering phase.

Loss function:

$$\mathcal{L} = \mathcal{L}_{\mathrm{R}} + \lambda \cdot \mathcal{L}_{\mathrm{C}}$$

Reconstruction Loss $$\mathcal{L}_{\mathrm{R}} = \sum_{x \in \mathcal{D}} \left|\left| \mathbf{x} - \mathrm{dec}\big(\mathrm{enc}(\mathbf{x})\big) \right|\right|_2^2$$

Compression Loss $$\mathcal{L}_{\mathrm{C}} = KL(P||Q)$$

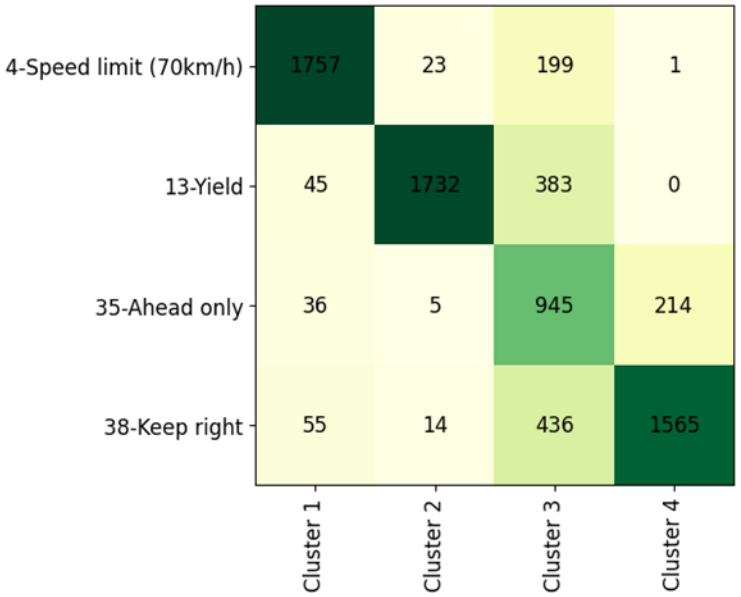
Conflicting goals

Compression - Abstraction: Learn a representation where the clusters collapse
Reconstruction - Representation: Preserve all information of every single object

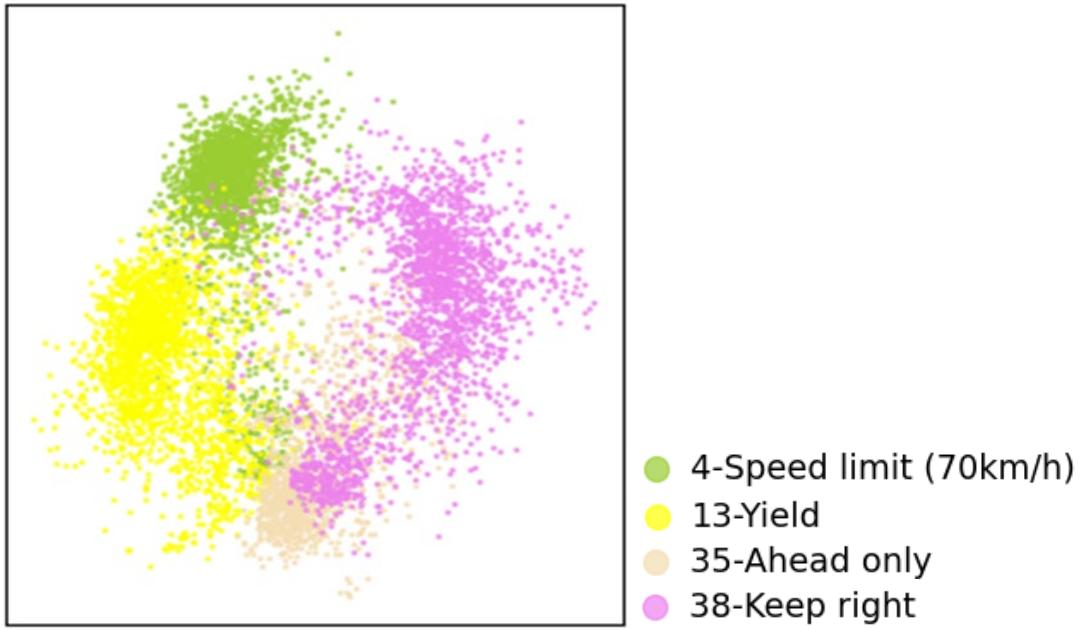
Guo, et al: Improved Deep Embedded Clustering with Local Structure Preservation. IJCAI 2017

IDEC

NMI 0.60



Cluster centers.



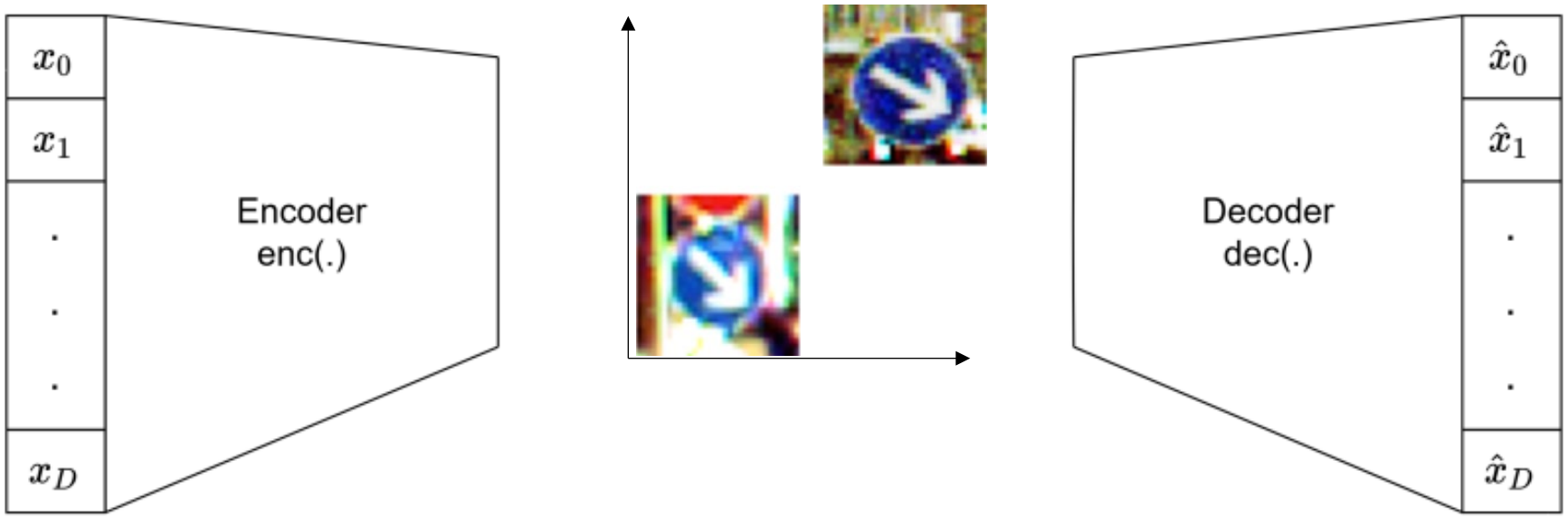
PCA of latent space to 2D.

IDEC: Original and Reconstructed Images

Original:

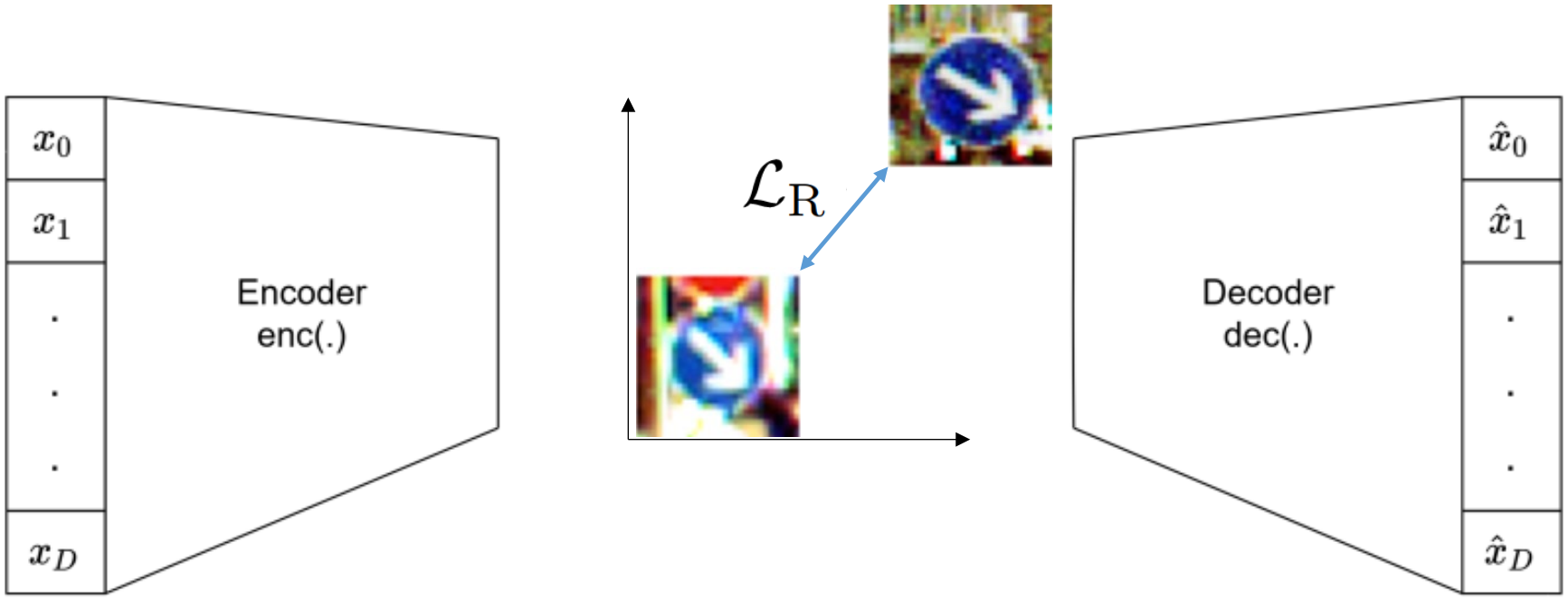
Reconstructed:

Abstraction and Representation as Conflicting Goals



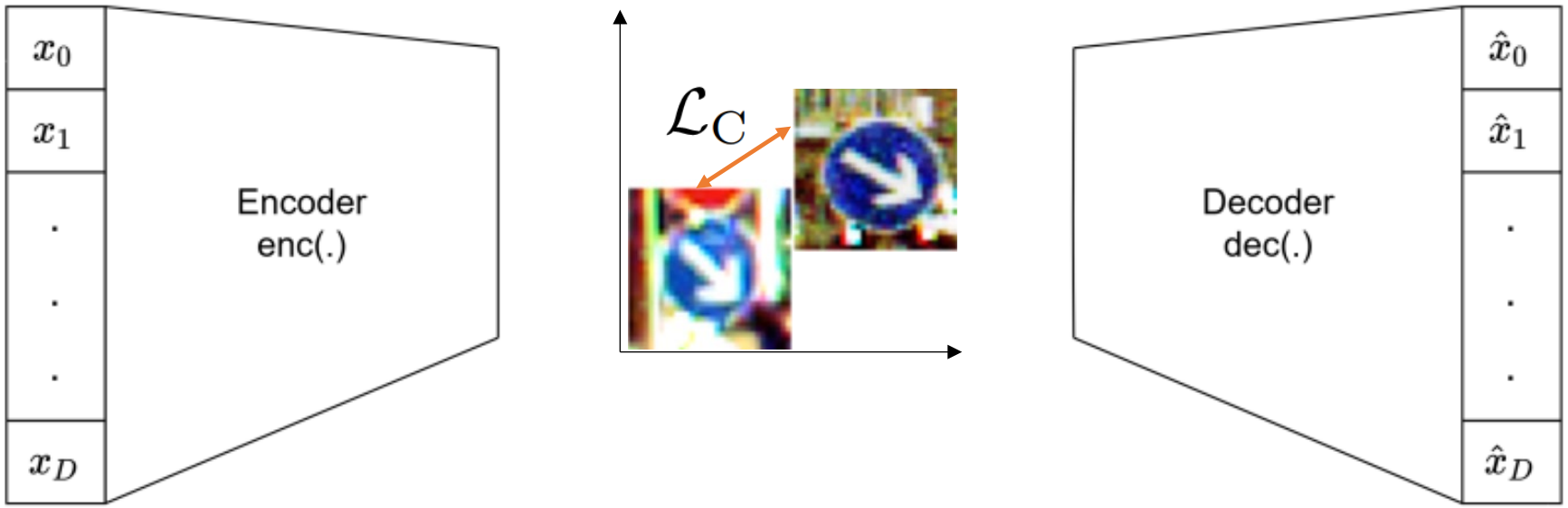
$$\mathcal{L} = \mathcal{L}_{\mathrm{R}} + \lambda \cdot \mathcal{L}_{\mathrm{C}}$$

Abstraction and Representation as Conflicting Goals



$$\mathcal{L} = \mathcal{L}_{\mathrm{R}} + \lambda \cdot \mathcal{L}_{\mathrm{C}}$$

Abstraction and Representation as Conflicting Goals



$$\mathcal{L} = \mathcal{L}_R + \lambda \cdot \mathcal{L}_C$$

Conflicting goals are difficult to optimize.

Solution 1)

Deep neural networks can learn difficult things, but they need to be encouraged to do so.

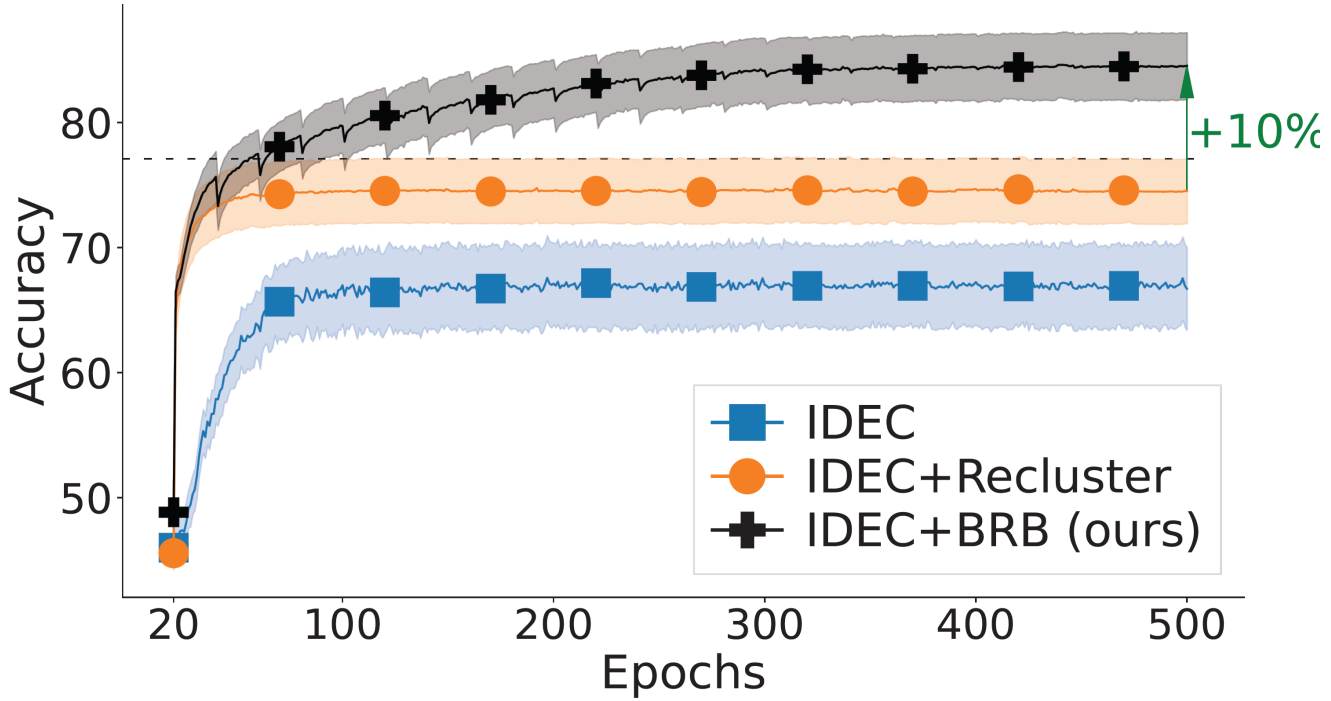
Solution 2)

Avoid the conflict.

Maintaining Neural Plasticity during Clustering

- **By weight resets**
- After each reset:

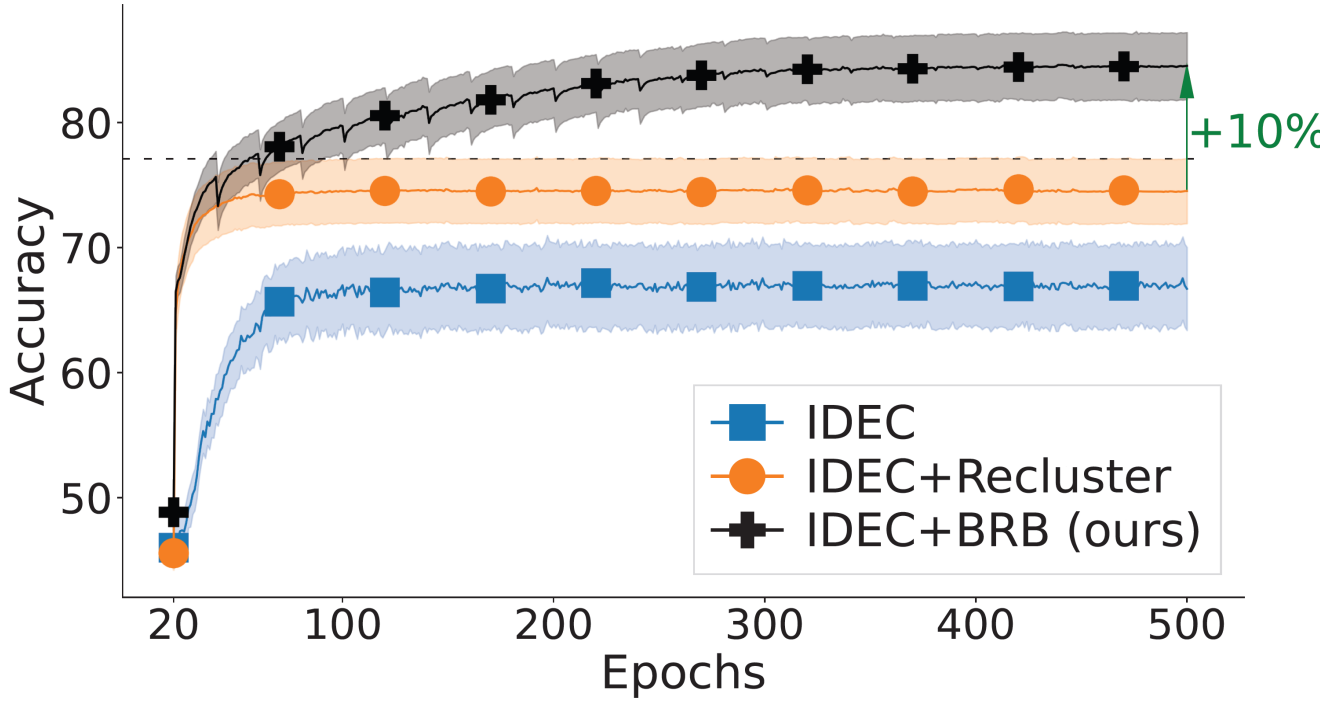
  reclustering with K-Means



L. Miklautz et al.: Breaking the Reclustering Barrier in Centroid-based Deep Clustering. ICLR 2025

Maintaining Neural Plasticity during Clustering

- **By weight resets**
- After each reset:
  reclustering with K-Means



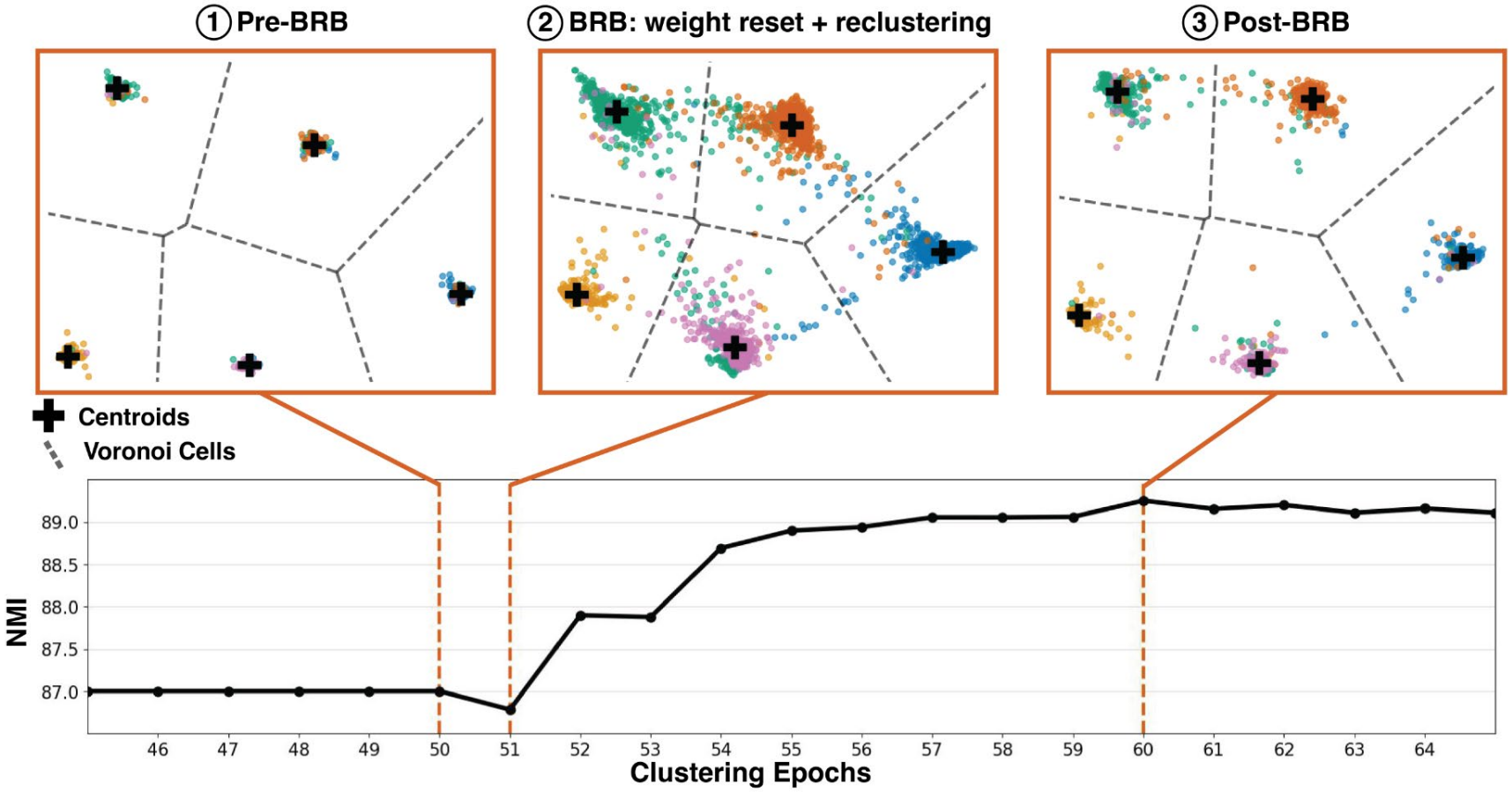
Weight resets are common in reinforcement learning to adapt to changing targets.

L. Miklautz et al.: Breaking the Reclustering Barrier in Centroid-based Deep Clustering. ICLR 2025

Changing Learning Targets in Deep Clustering

1) **Pre-training – only focus on representation:** Use an autoencoder to learn a non-linear embedding of your data i.e., Feature learning/Representation learning
2) **Obtain initial cluster labels**; typically with K-Means
3) **Deep clustering – mostly focus on abstraction:** Refine the embedding and the clustering

Breaking the Reclustering Barrier

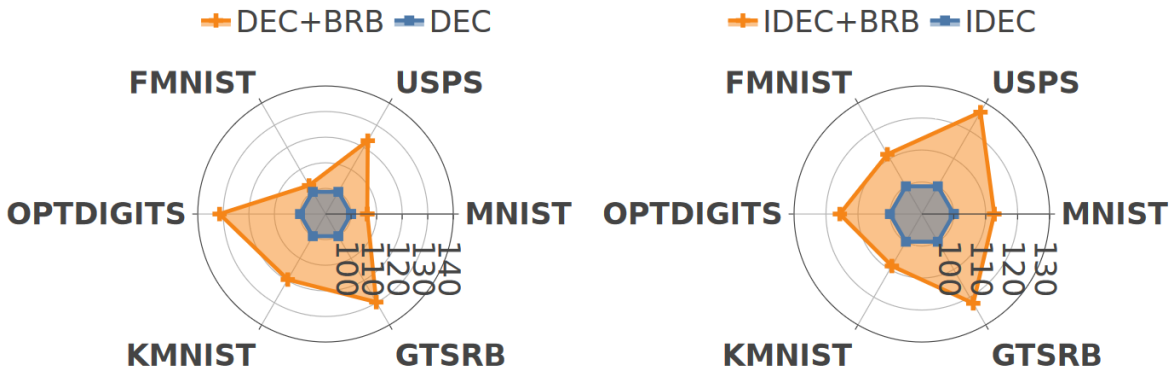


DEC + BRB on USPS Data.

Ablation Studies and Experimental Results

- BRB prevents early over-commitment by increasing intra-cluster variance while preserving cluster separation.
- Enables training from scratch
- Performance improvements in almost all settings, also with contrastive pretraining
- Runtime overhead about 1%

Training from scratch:
Relative improvements of clustering accuracy.



Conflicting goals are difficult to optimize.

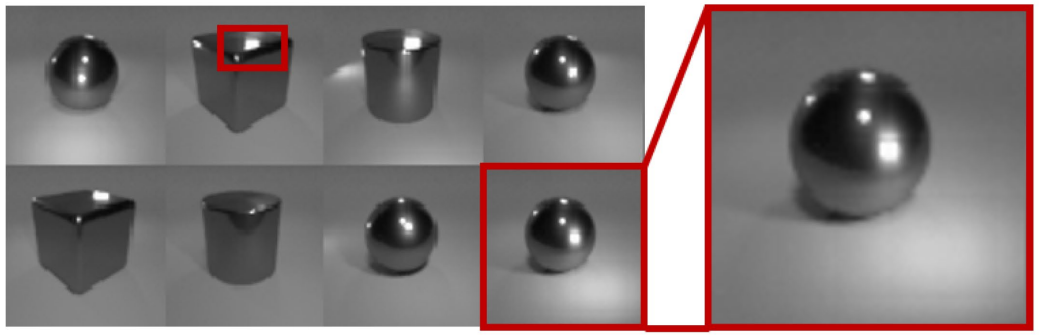
Solution 1)

Deep neural networks can learn difficult things, but they need to be encouraged to do so.

Solution 2)

Avoid the conflict.

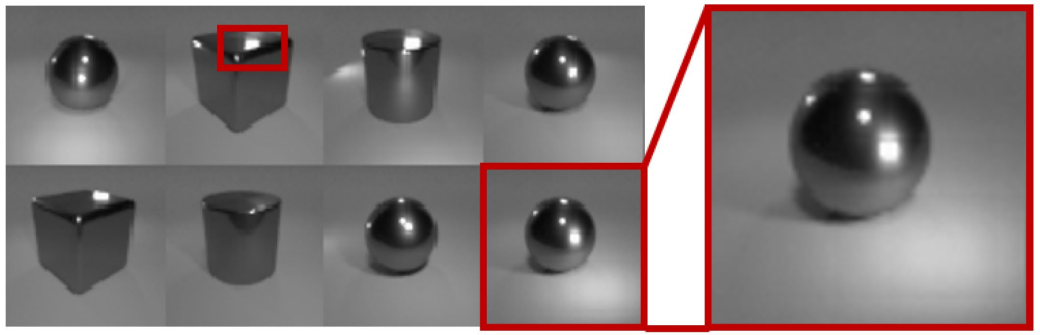
Avoid the conflict by learning two representations.



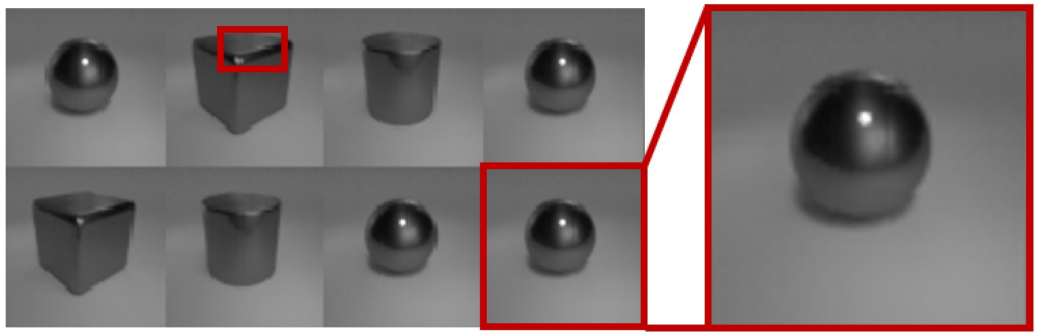
Input Data

L. Miklautz et al.: Details (Don’t) Matter: Isolating Cluster Information in Deep Embedded Spaces. IJCAI 2021

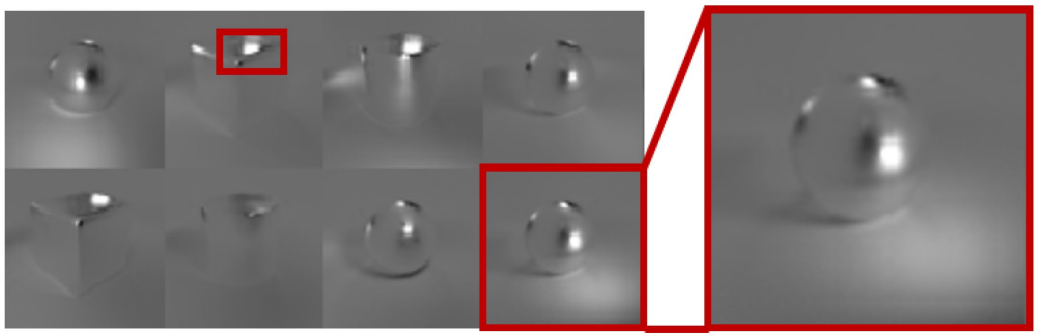
Avoid the conflict by learning two representations.



Input Data



Clustered Space Focussing on Abstraction



Shared Space Focussing on Representation

L. Miklautz et al.: Details (Don't) Matter: Isolating Cluster Information in Deep Embedded Spaces. IJCAI 2021

Framework for Isolating Cluster Information

in **A**utoencoder **Ce**ntroid-based **De**ep **C**lustering methods (ACe/DeC)

Objective Function:

$$\mathcal{J} = \underbrace{\left[\sum_{i=1}^{k}\sum_{\mathbf{x}\in C_i}\left|\left|V^{\mathsf{T}}\mathrm{enc}(\mathbf{x}) - V^{\mathsf{T}}\mu_i\right|\right|_{\beta_{cs}}^{2}\right]}_{\text{Clustered Space}} + \underbrace{\left[\sum_{\mathbf{x}\in\mathcal{D}}\left|\left|V^{\mathsf{T}}\mathrm{enc}(\mathbf{x}) - V^{\mathsf{T}}\mu_{\mathcal{D}}\right|\right|_{\beta_{ss}}^{2}\right]}_{\text{Shared Space}} + \underbrace{\left[\sum_{\mathbf{x}\in\mathcal{D}}\left|\left|\mathbf{x} - \mathrm{dec}\big(VV^{\mathsf{T}}\mathrm{enc}(\mathbf{x})\big)\right|\right|_{2}^{2}\right]}_{\text{Reconstruction Error}}$$

Differences to Sub-K-Means

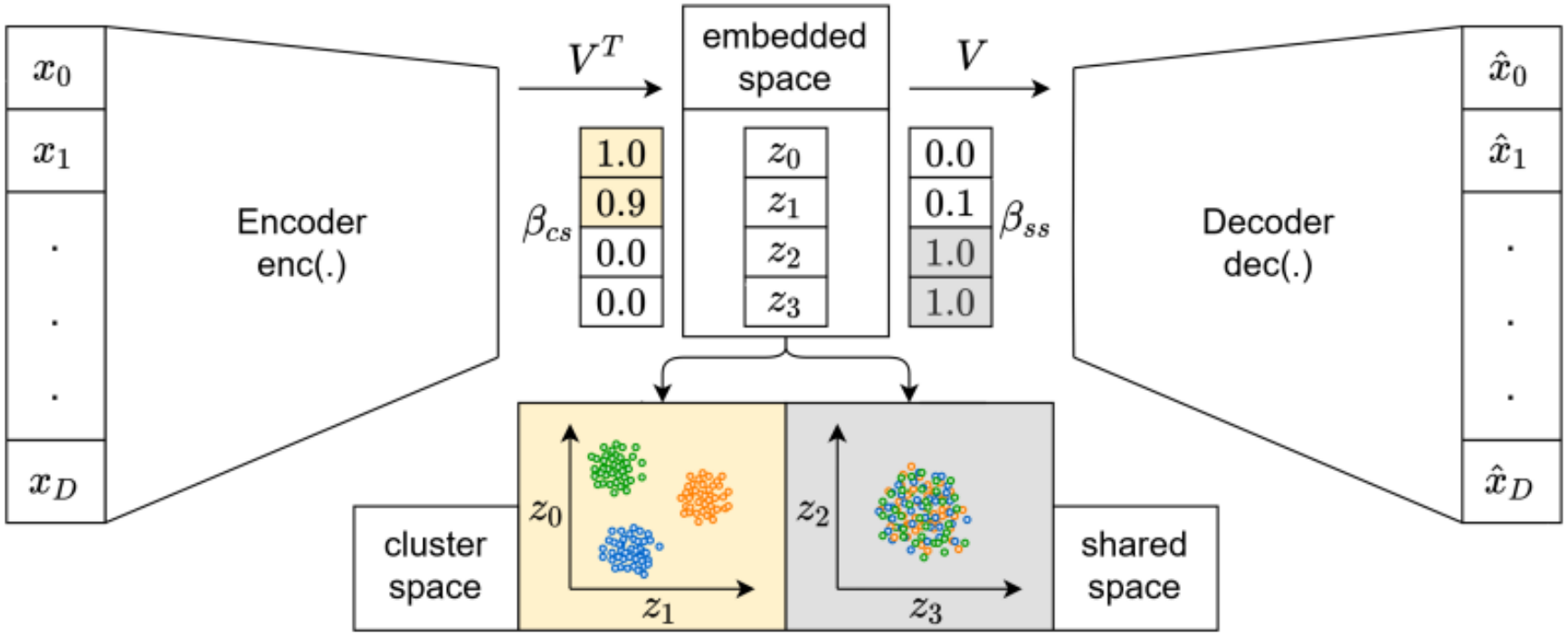
in **A**utoencoder **Ce**ntroid-based **De**ep **C**lustering methods (ACe/DeC)

Objective Function:

$$\mathcal{J} = \underbrace{\left[\sum_{i=1}^{k}\sum_{\mathbf{x}\in C_i}\left|\left|V^{\mathsf{T}}\mathrm{enc}(\mathbf{x}) - V^{\mathsf{T}}\mu_i\right|\right|^2_{\beta_{cs}}\right]}_{\text{Clustered Space}} + \underbrace{\left[\sum_{\mathbf{x}\in\mathcal{D}}\left|\left|V^{\mathsf{T}}\mathrm{enc}(\mathbf{x}) - V^{\mathsf{T}}\mu_{\mathcal{D}}\right|\right|^2_{\beta_{ss}}\right]}_{\text{Shared Space}} + \underbrace{\left[\sum_{\mathbf{x}\in\mathcal{D}}\left|\left|\mathbf{x} - \mathrm{dec}\left(VV^{\mathsf{T}}\mathrm{enc}(\mathbf{x})\right)\right|\right|^2_2\right]}_{\text{Reconstruction Error}}$$

- **Assignment matrix V is not necessarily orthogonal;**
- **β are trainable weights constrained by a sigmoid function**

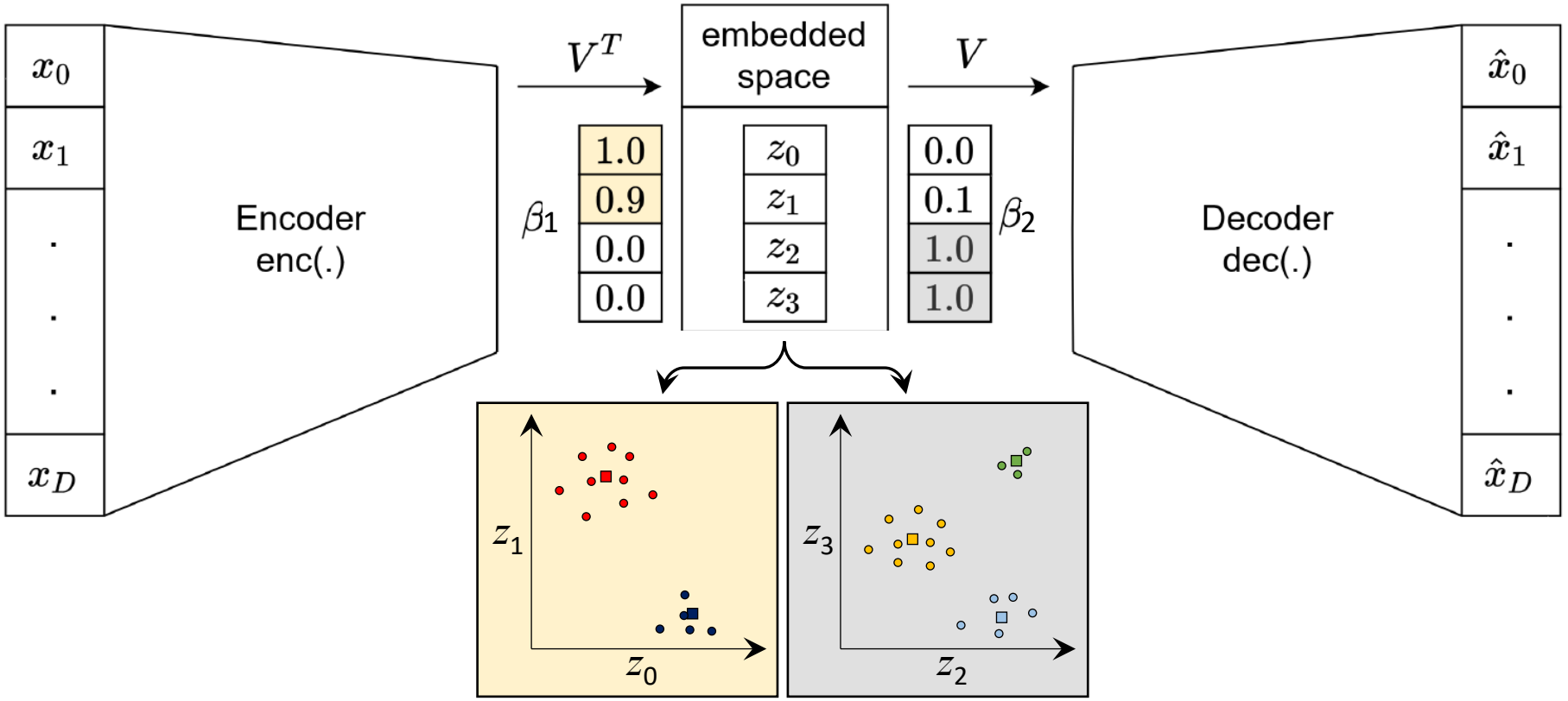
Isolating Cluster Information



Framework ACe/DeC

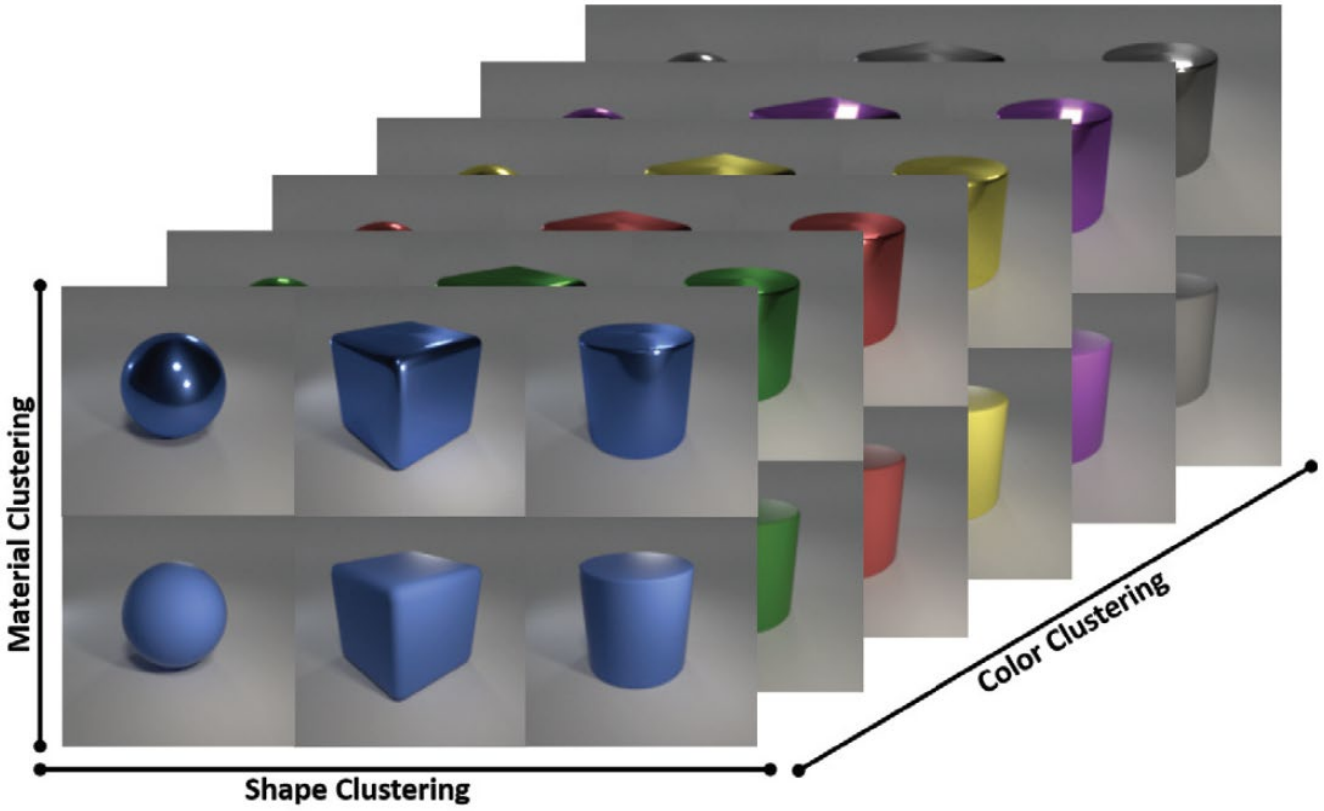
1) **Pre-training:** Use an autoencoder to learn a non-linear embedding of your data
2) **Initalization:** initialize V as a random orthonormal matrix, randomly assign each dimension to Clustered/Shared Space; perform K-Means in the Clustered Space; Keep the autoencoder parameters fixed and optimize V, β
3) **Update** all parameters in minibatch training: cluster labels with the underlying deep clustering algorithm, autoencoder parameters, V, β

Finding Multiple Alternative Clusterings



Miklautz et al.: Deep Embedded Non-Redundant Clustering. AAAI 2020

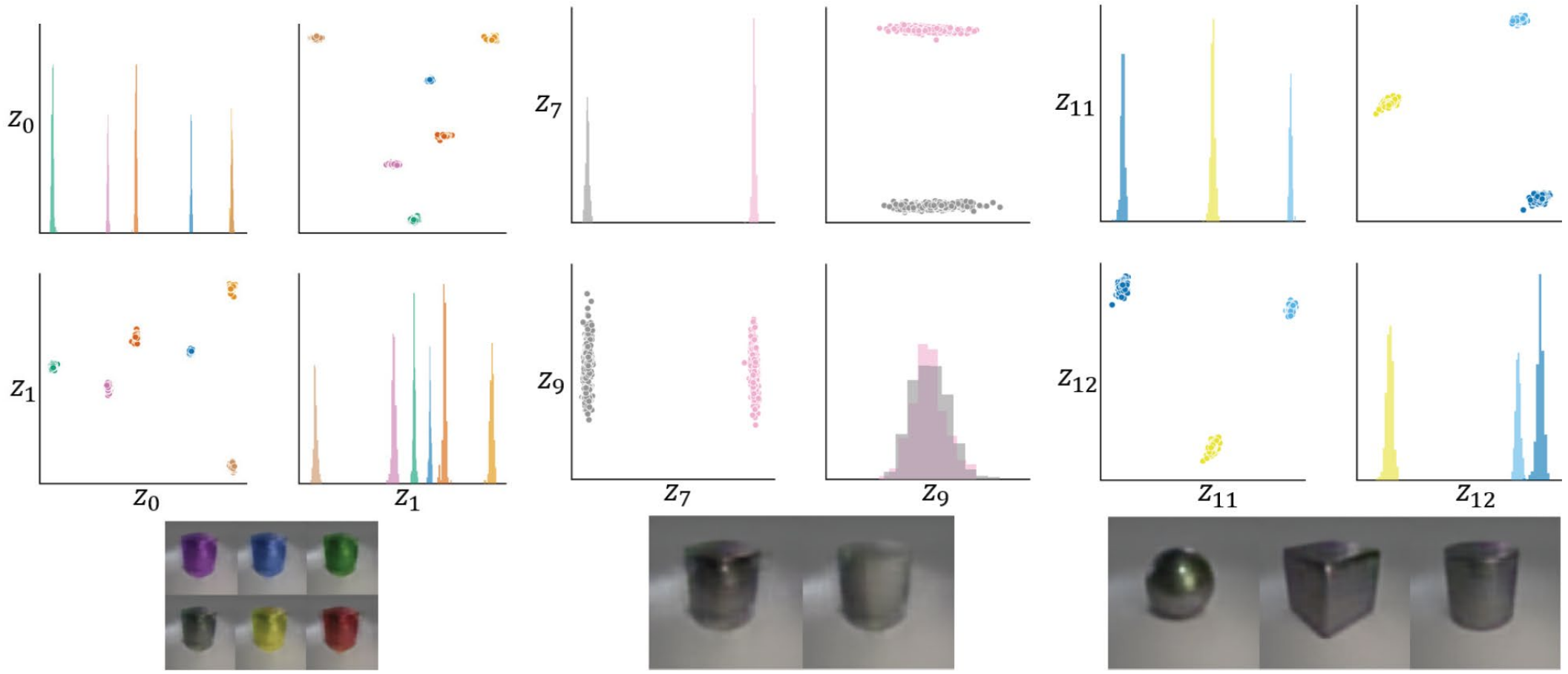
ENRC - Experiments



NR-Objects data

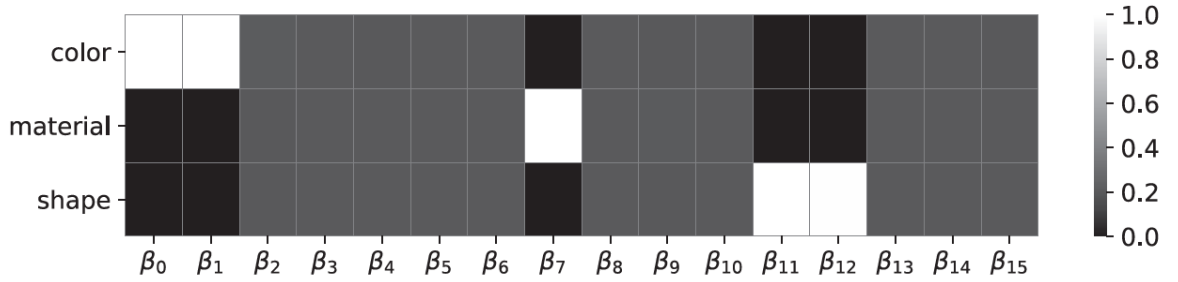
16,384 dimensions
10,000 objects

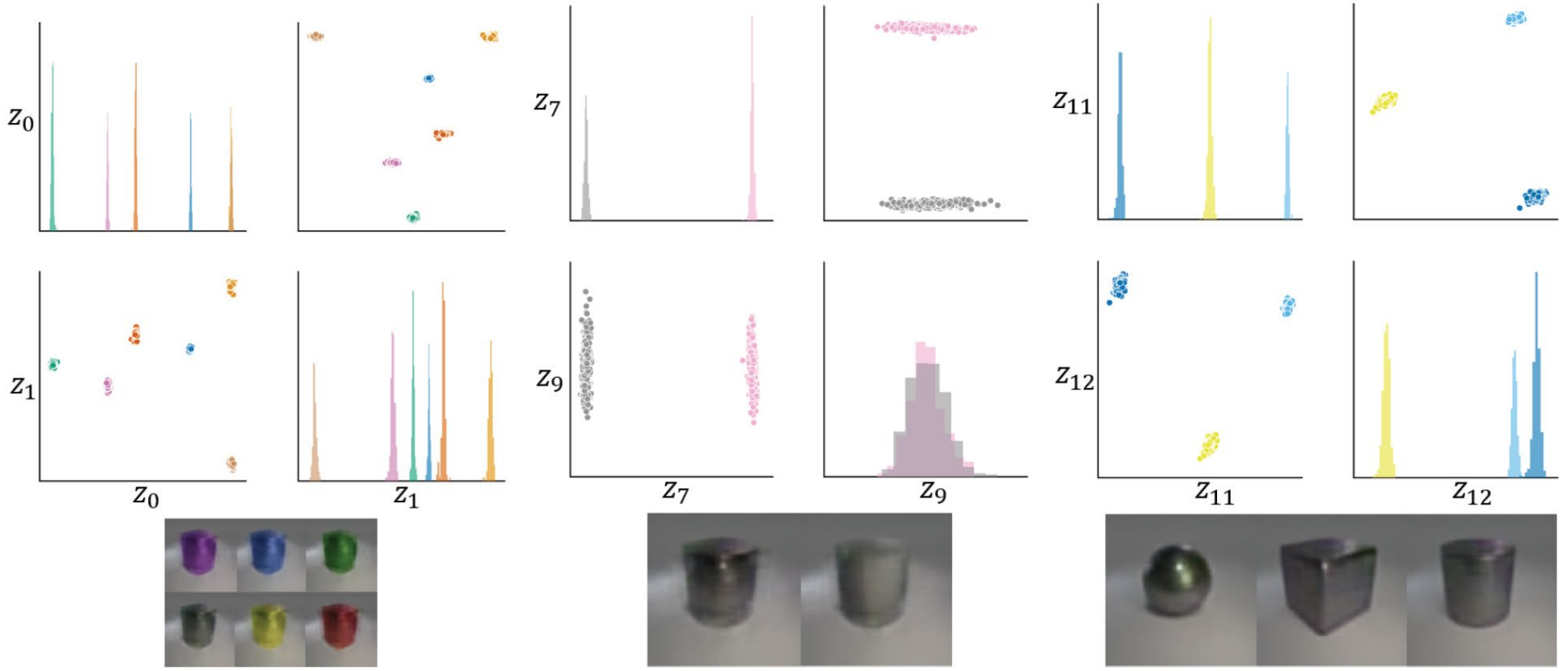
ENRC - Experiments



(a) Color (b) Material (c) Shape

ENRC - Experiments





(a) Color (b) Material (c) Shape

ENRC - Experiments

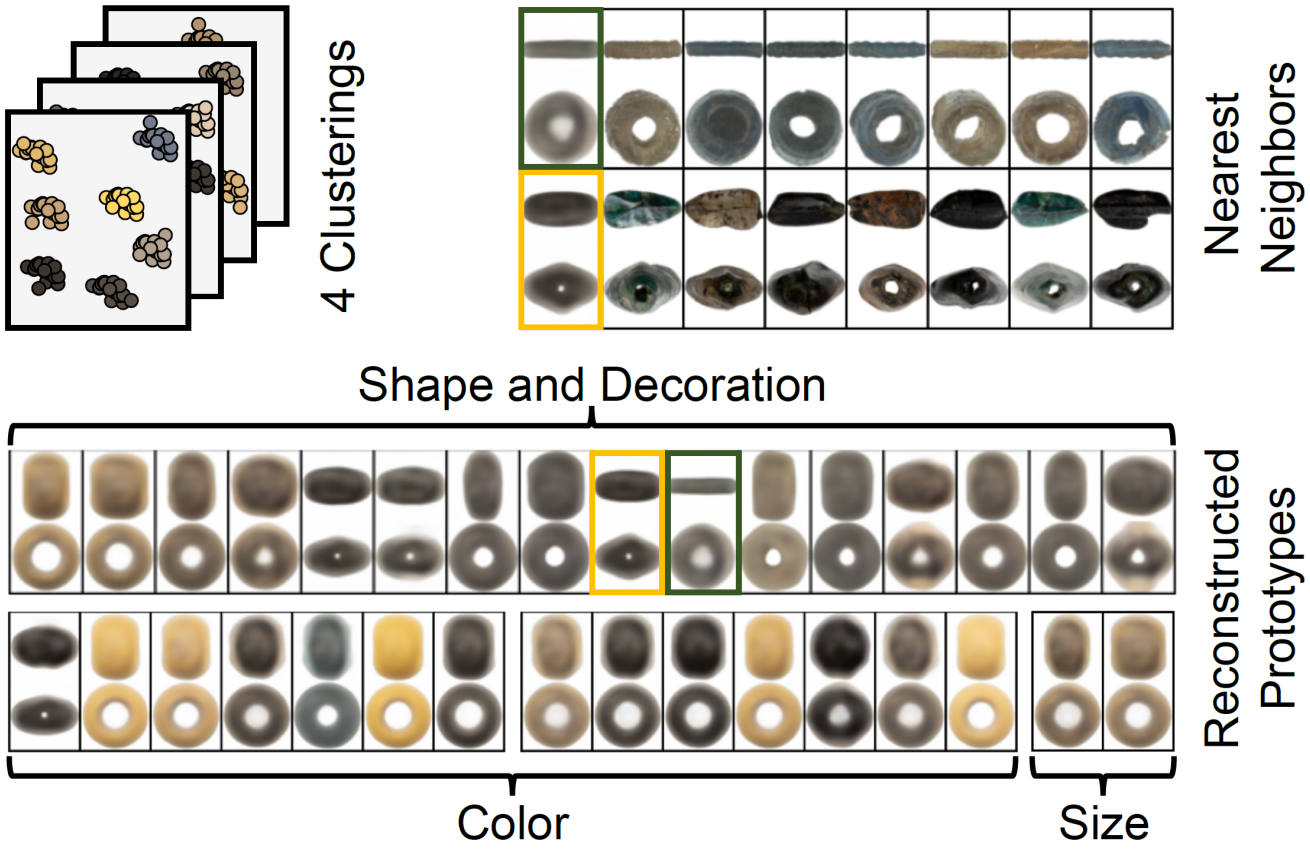
Data Sets	Clustering	ENRC	Orth1	Orth2	mSC	Nr-Kmeans	ISAAC
NR-Objects	color	**1.00 ±0.00**	0.70 ±0.09	0.73 ±0.06	0.35 ±0.05	0.92 ±0.09	0.15 ±0.06
	material	**1.00 ±0.00**	0.46 ±0.16	0.11 ±0.12	0.03 ±0.07	0.95 ±0.14	0.53 ±0.08
	shape	**1.00 ±0.00**	0.39 ±0.20	0.20 ±0.08	0.03 ±0.03	0.92 ±0.16	0.60 ±0.07
GTSRB	type	**0.74 ±0.01**	0.57 ±0.07	0.73 ±0.15	0.04 ±0.04	0.72 ±0.01	0.60 ±0.07
	color	**0.67 ±0.00**	0.59 ±0.02	0.63 ±0.03	0.04 ±0.06	0.65 ±0.01	0.59 ±0.04
Stickfigures	upper	**1.00 ±0.00**	0.79 ±0.21	0.00 ±0.00	0.33 ±0.20	**1.00 ±0.00**	0.37 ±0.05
	lower	**1.00 ±0.00**	0.77 ±0.24	0.00 ±0.00	0.30 ±0.17	**1.00 ±0.00**	0.39 ±0.08
C-MNIST	left	**0.83 ±0.04**	0.33 ±0.02	0.35 ±0.03	0.07 ±0.02*	0.69 ±0.03	0.29 ±0.13*
	right	**0.82 ±0.01**	0.40 ±0.03	0.41 ±0.04	0.06 ±0.02*	0.70 ±0.03	0.19 ±0.13*

type

color

German Traffic Sign Benchmark Data

Application: Medieval Glass Beads from Vienna



Outliers

Miklautz et al.. Non-redundant Image Clustering of Early Medieval Glass Beads. DSAA 2023

Conflicting goals are difficult to optimize.

Solution 1)

Deep neural networks can learn difficult things, but they need to be encouraged to do so.

Solution 2)

Avoid the conflict.

Avoid the conflict by local, cluster-wise abstraction.

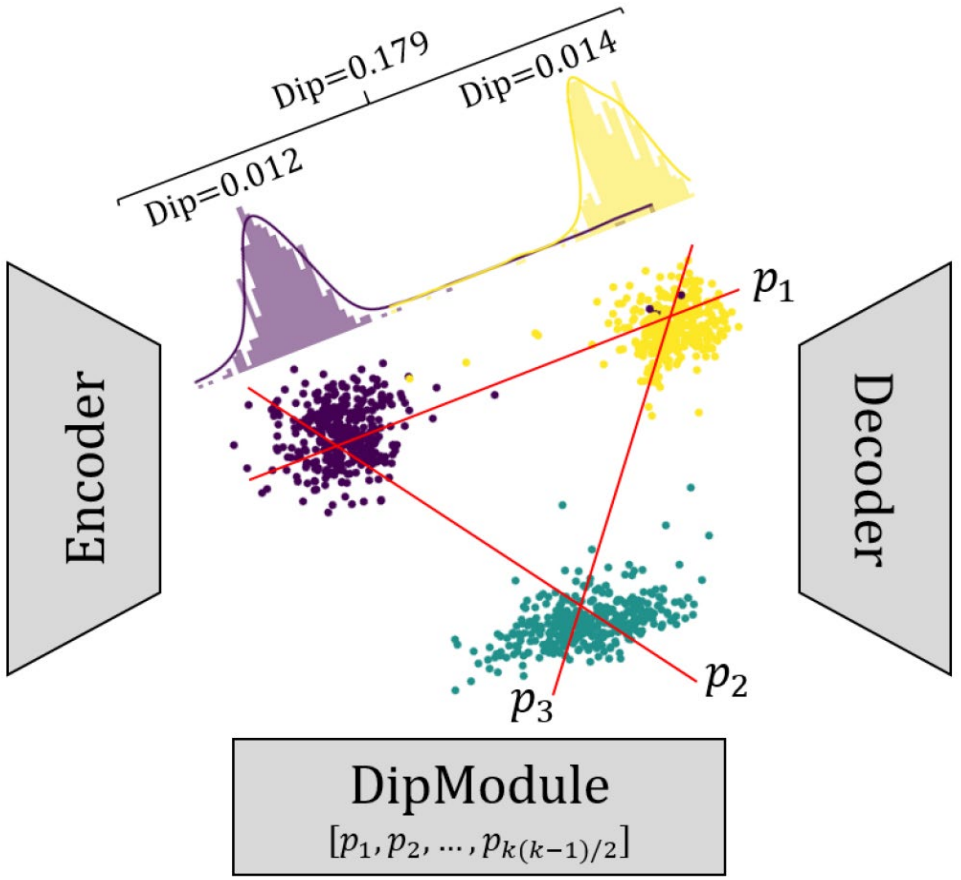
DipEncoder integrates the
Dip-test for multimodality in an autoencoder.

For **each pair of clusters** the DipModule learns:

- The projection on the direction of maximal bi-modality,
- Maximizes Dip-value on this projection
- Mimizes the Dip-value of each individual cluster.

$$\mathcal{L}_{dip}(\mathcal{B}) = \frac{2}{k(k-1)} \sum_{a=1}^{(k-1)} \sum_{b=a+1}^{k} \mathcal{L}_{uni}(\mathcal{B}, a, b) + \mathcal{L}_{multi}(\mathcal{B}, a, b)$$

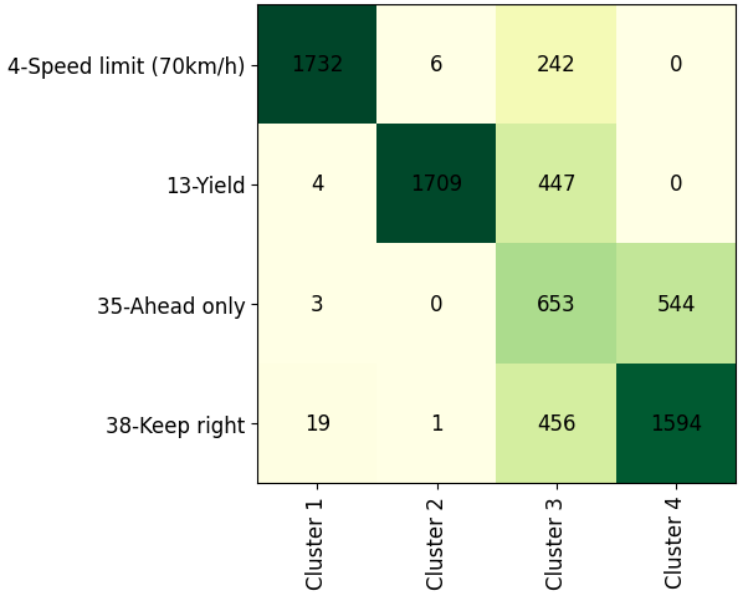
$$\mathcal{L}_{final}(\mathcal{B}) = \mathcal{L}_{dip}(\mathcal{B}) + \lambda \mathcal{L}_{rec}(\mathcal{B})$$



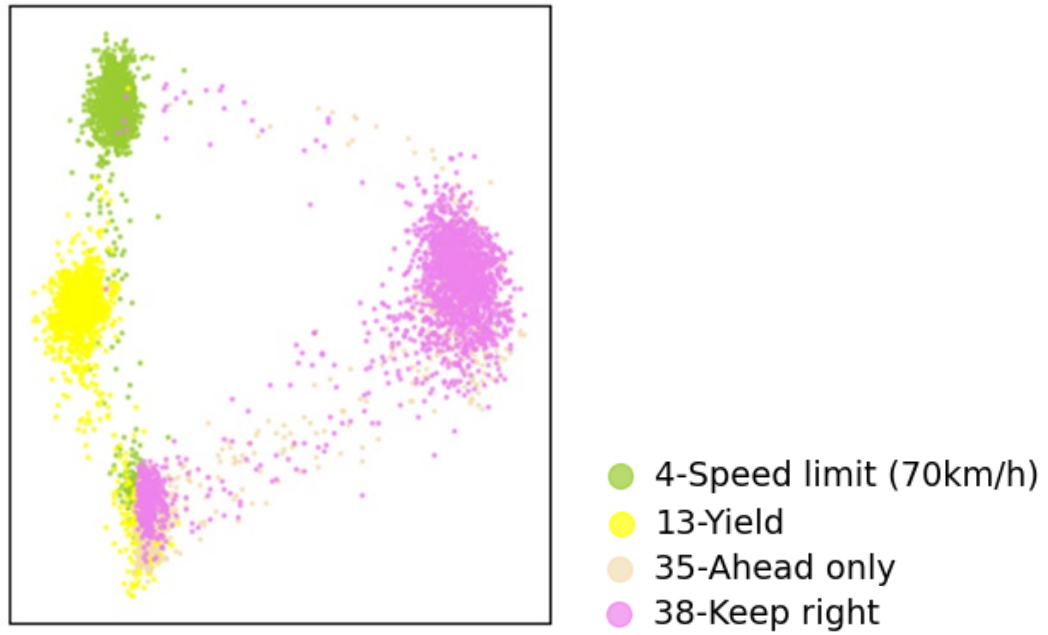
Leiber et al. The DipEncoder: Enforcing Multimodality in Autoencoders. KDD 2022

DipEncoder

NMI 0.62



Cluster centers.



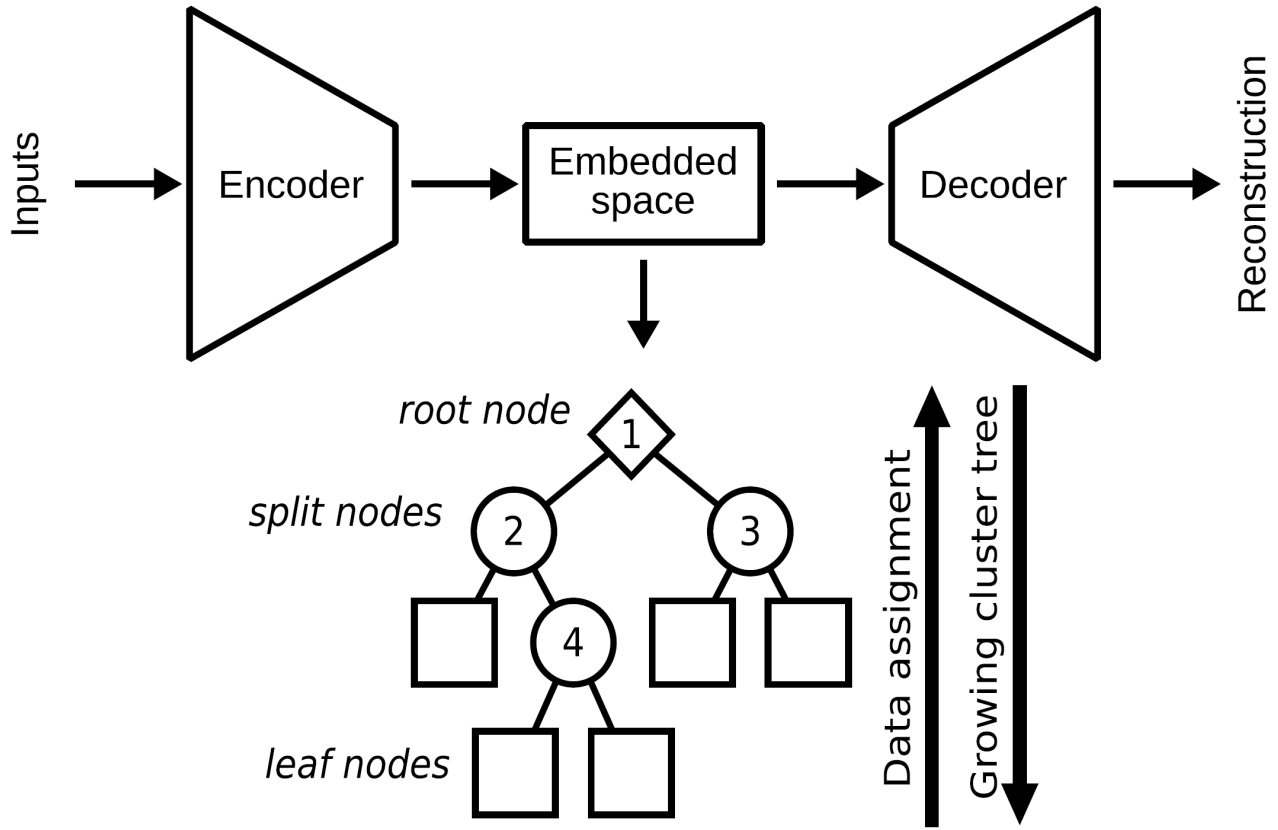
PCA of latent space to 2D.

DipEncoder: Original and Reconstructed Images

Original:

Reconstructed:

Avoid the conflict by hierarchical abstraction.



Mautz et al.: DeepECT: The Deep Embedded Cluster Tree. Data Sci. Eng. 5(4): 419-432 (2020)

Deep Embedded Cluster Tree

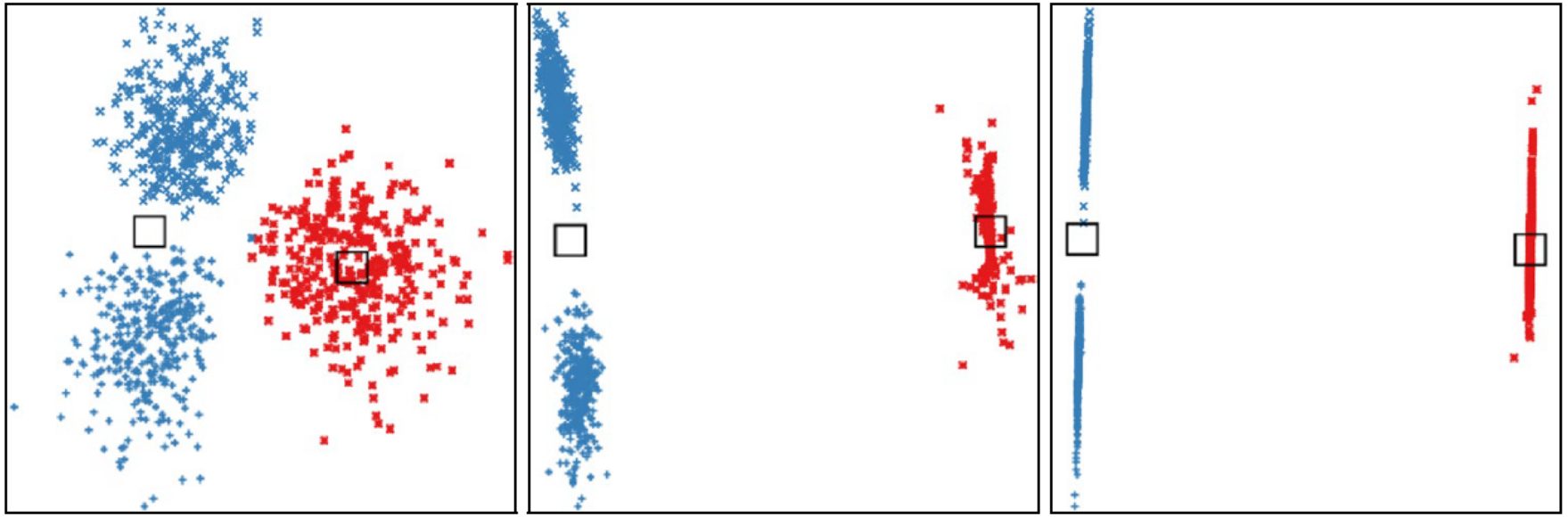
Objective Function

$$\mathcal{L} = \mathcal{L}_{\mathrm{DC}} + \mathcal{L}_{\mathrm{NC}} + \mathcal{L}_{\mathrm{R}}$$

$\mathcal{L}_{\mathrm{NC}}$ **Node Compression Loss:** Maximize squarred distance between leafs

$\mathcal{L}_{\mathrm{R}}$ Reconstruction error.

$\mathcal{L}_{\mathrm{DC}}$ **Node Data Compression Loss:** minimizes the distance of data point and center of a node projected on the connecting line with sibling node.



Deep Embedded Cluster Tree

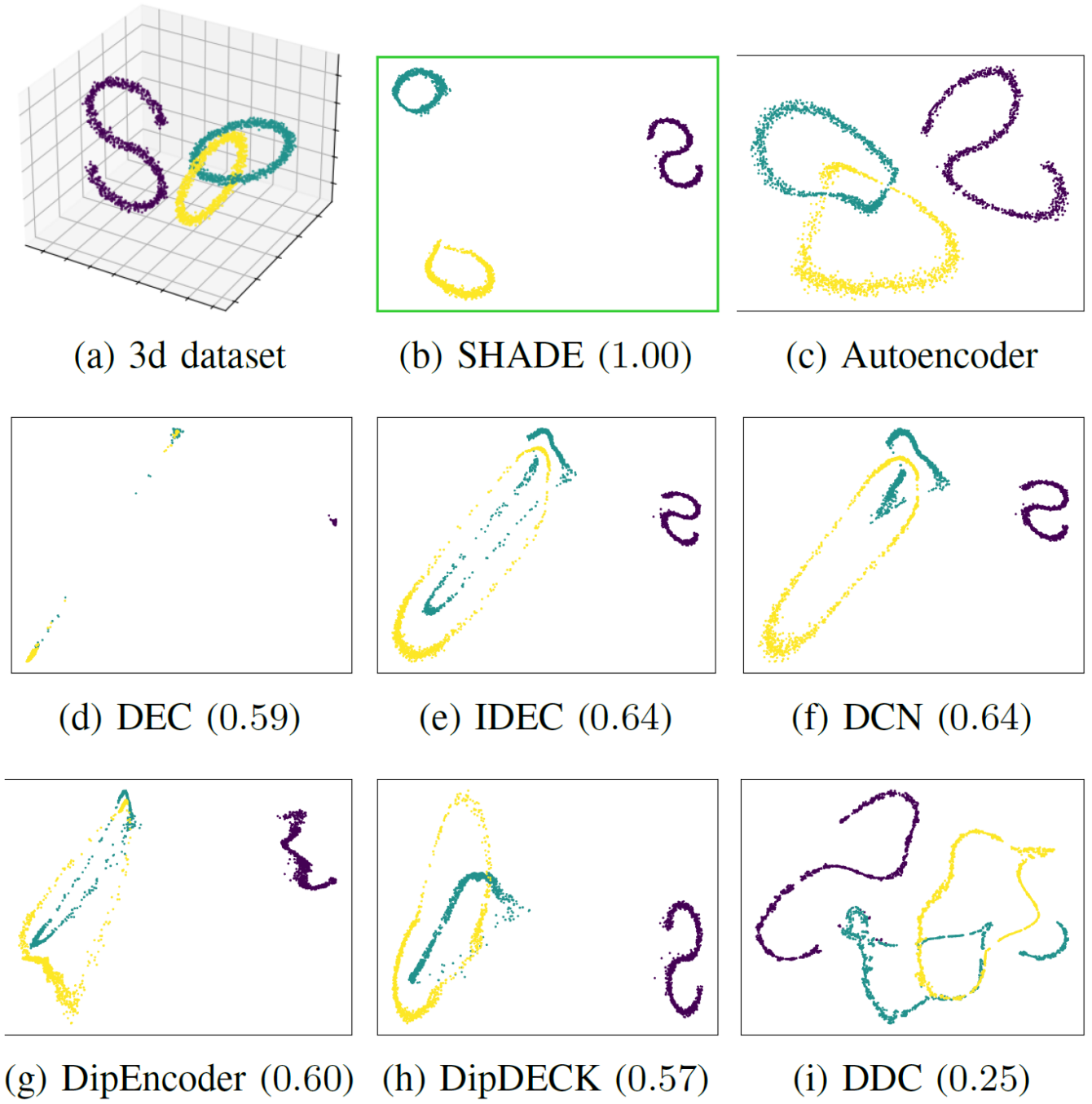
DeepECT: Original and Reconstructed Images

Original:

Reconstructed:

Density-based: Cluster order as guideline for embedding

- Algorithm SHADE computes the cluster order in original space
- Preserves this order in additon to autoencoder reconstruction error
- **Sequential deep density-based clustering method**



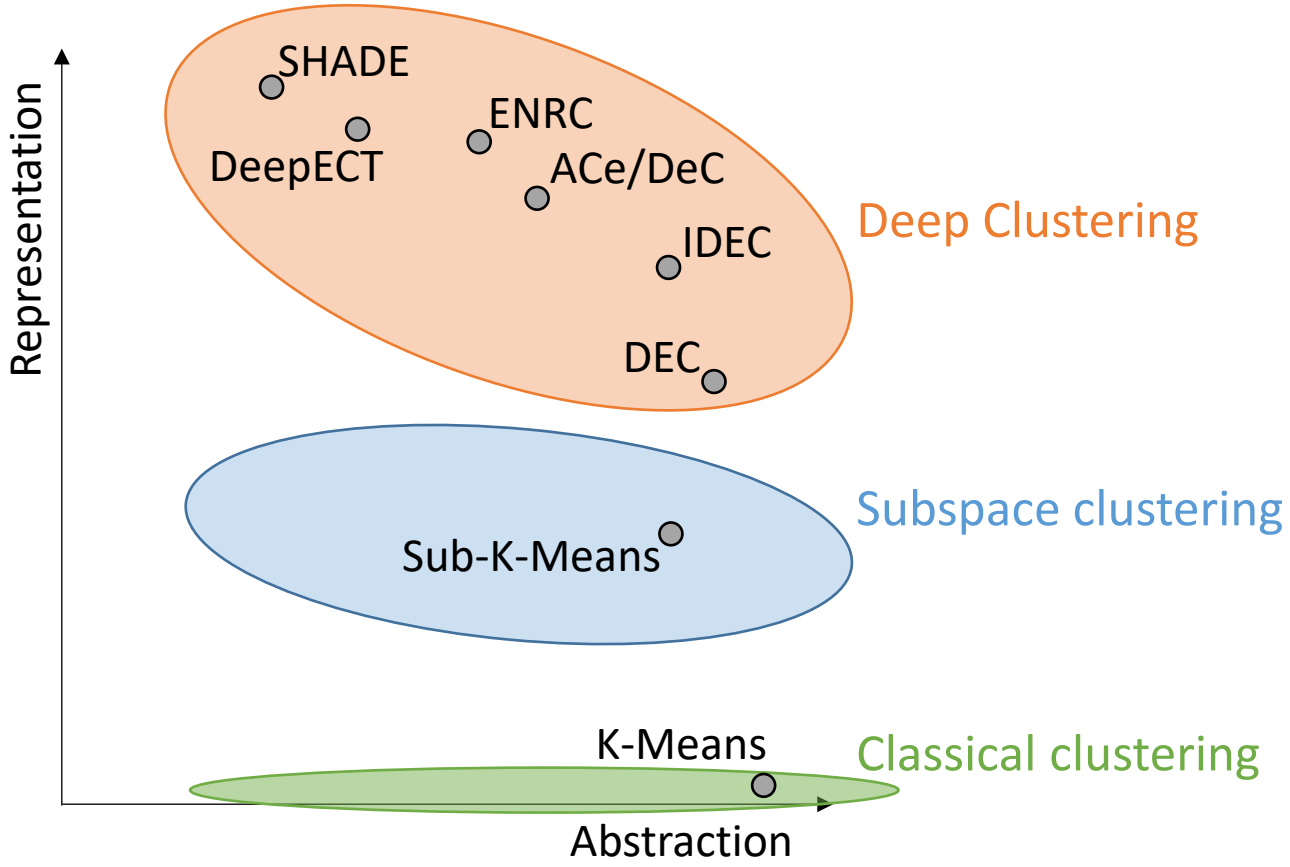
(a) 3d dataset (b) SHADE (1.00) (c) Autoencoder

(d) DEC (0.59) (e) IDEC (0.64) (f) DCN (0.64)

(g) DipEncoder (0.60) (h) DipDECK (0.57) (i) DDC (0.25)

Beer et al.: Deep Density-based Clustering. ICDM 2024

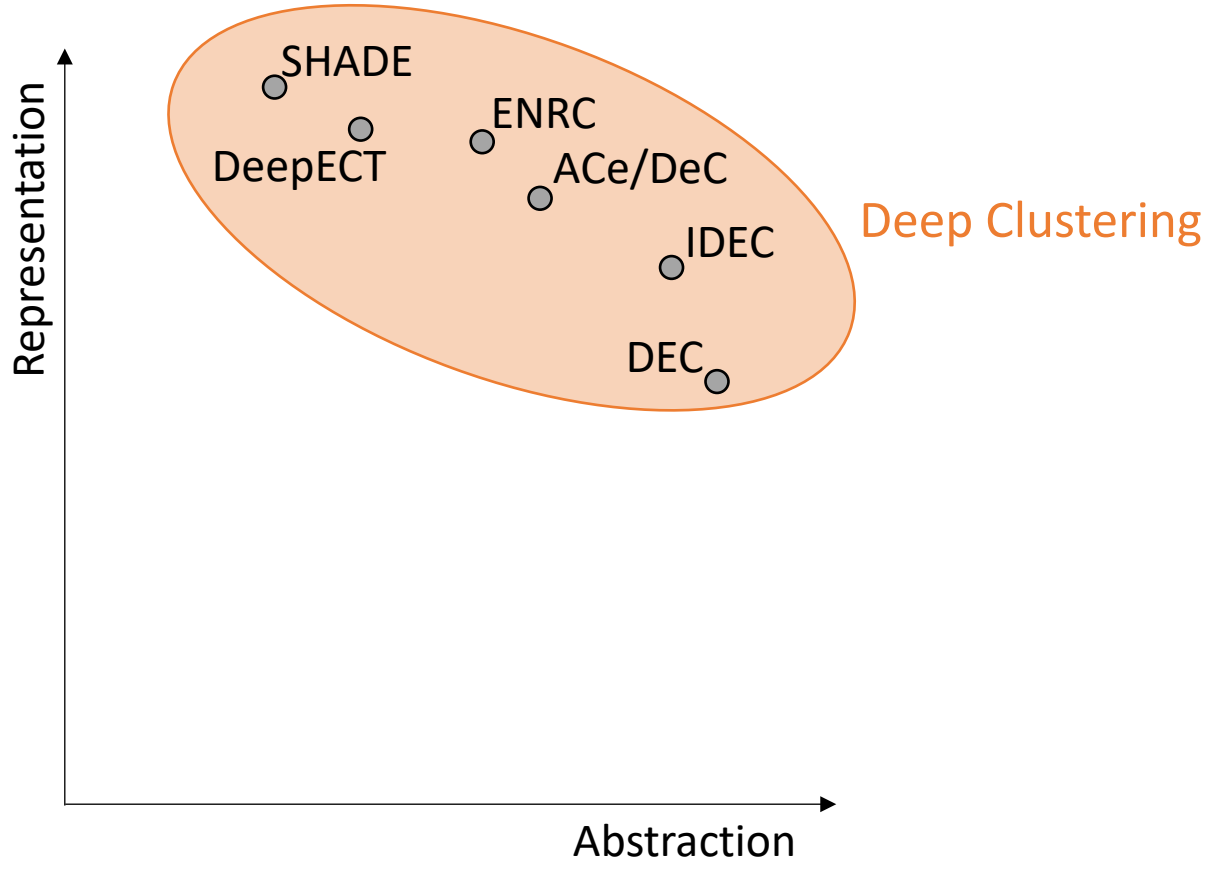
Balancing Abstraction and Representation



High representational complexity:
+ captures details
+ high accuracy
– difficult to interpret
+ little information loss

High level of abstraction:
+ generalizes
+ removes noise
+ easy to interpret
– high information loss

A Taxonomy of Deep Clustering Methods



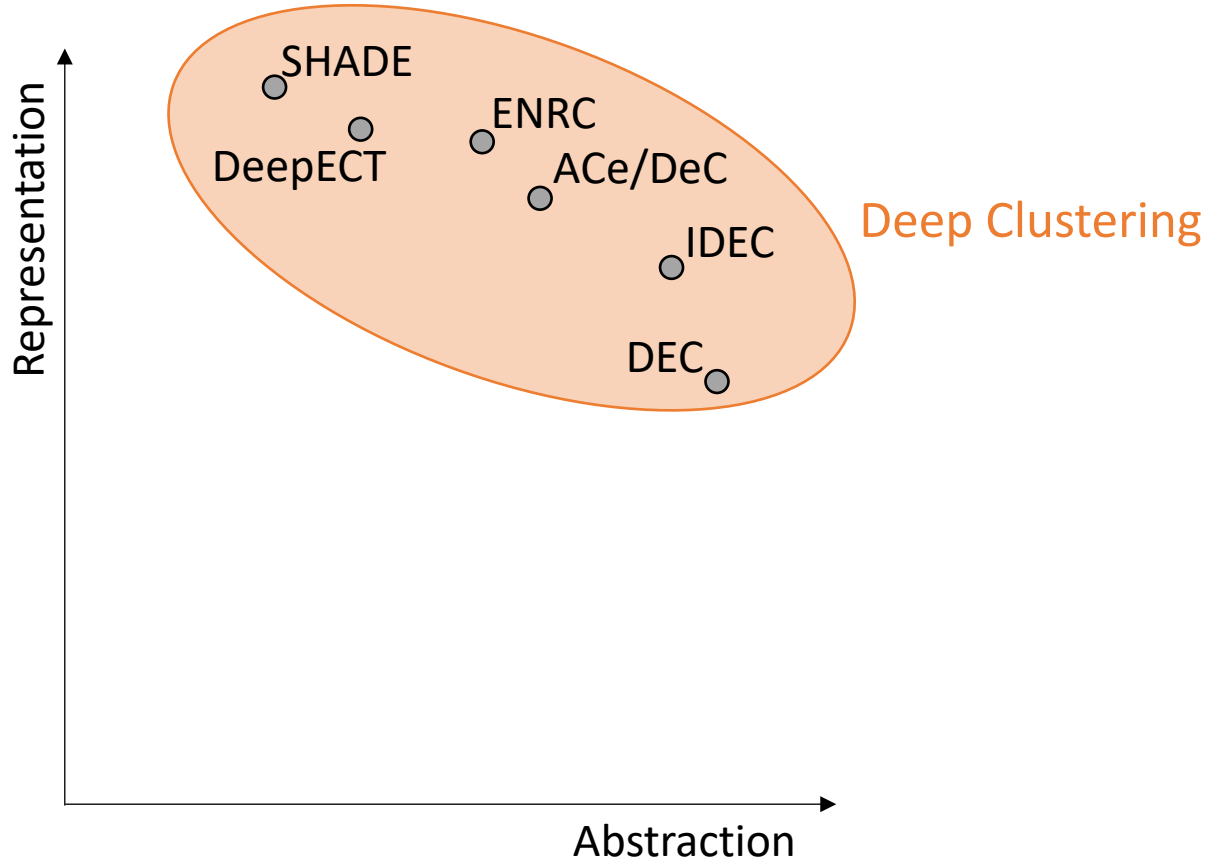
Repesentation Learning:
- autoencoder
- contrastive
- generative

Clustering:
- centroid-based
- hierarchical, density-based, spectral

Integration of RL and CL:
- sequential
- iterative
- simultaneous

Zhou et al.: A Comprehensive Survey on Deep Clustering: Taxonomy, Challenges, and Future Directions. ACM Comput. Surv. 2025

A Taxanomy of Deep Clustering Methods



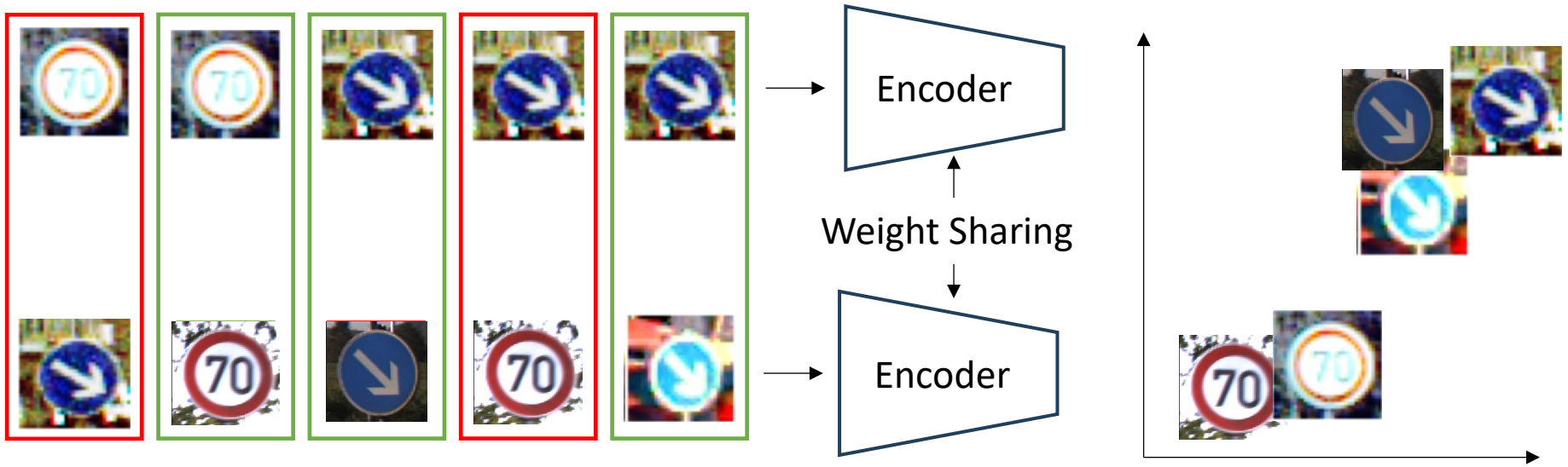
Repesentation Learning:
- autoencoder
- **contrastive**
- **generative**

Clustering:
- centroid-based
- hierarchical, density-based, spectral

Integration of RL and CL:
- sequential
- iterative
- simultaneous

Zhou et al.: A Comprehensive Survey on Deep Clustering: Taxonomy, Challenges, and Future Directions. ACM Comput. Surv. 2025

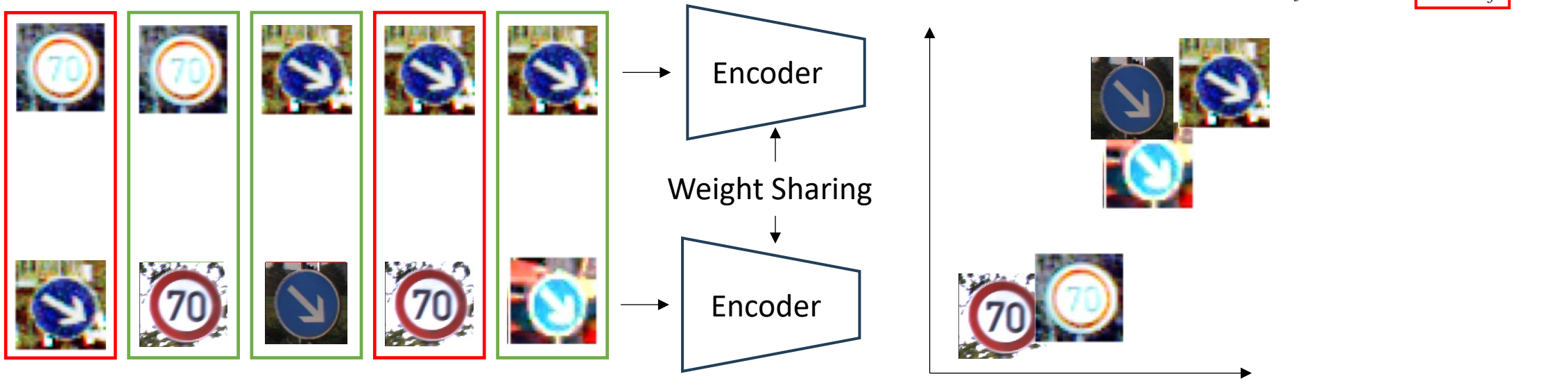
Contrastive Representation Learning



- Given positive and negative examples it is possible to learn high-quality representations.
- Labeled data supports contructing effective examples.

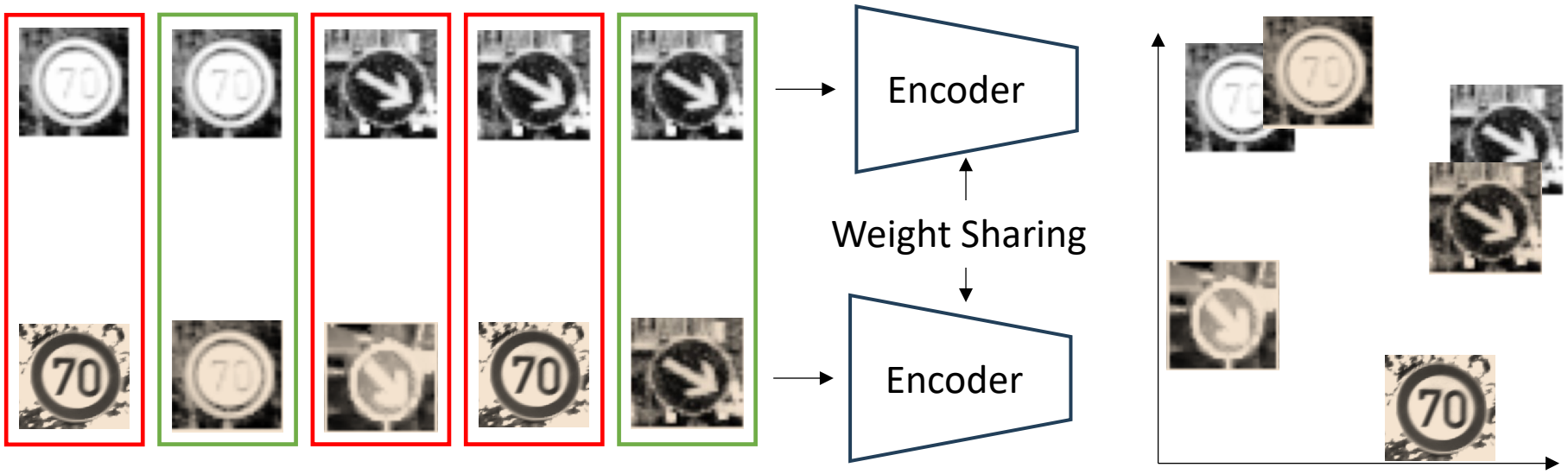
Contrastive Representation Learning

$$\mathcal{L}_{InfoNCE} = -\frac{1}{|N|}\sum_{i=1}^{N}\log\frac{\exp(f(h_i^{\mathcal{T}_1}, h_i^{\mathcal{T}_2})/\tau)}{\sum_{j=1}^{N}\exp(f(h_i^{\mathcal{T}_1}, h_j^{\mathcal{T}_2})/\tau)}$$



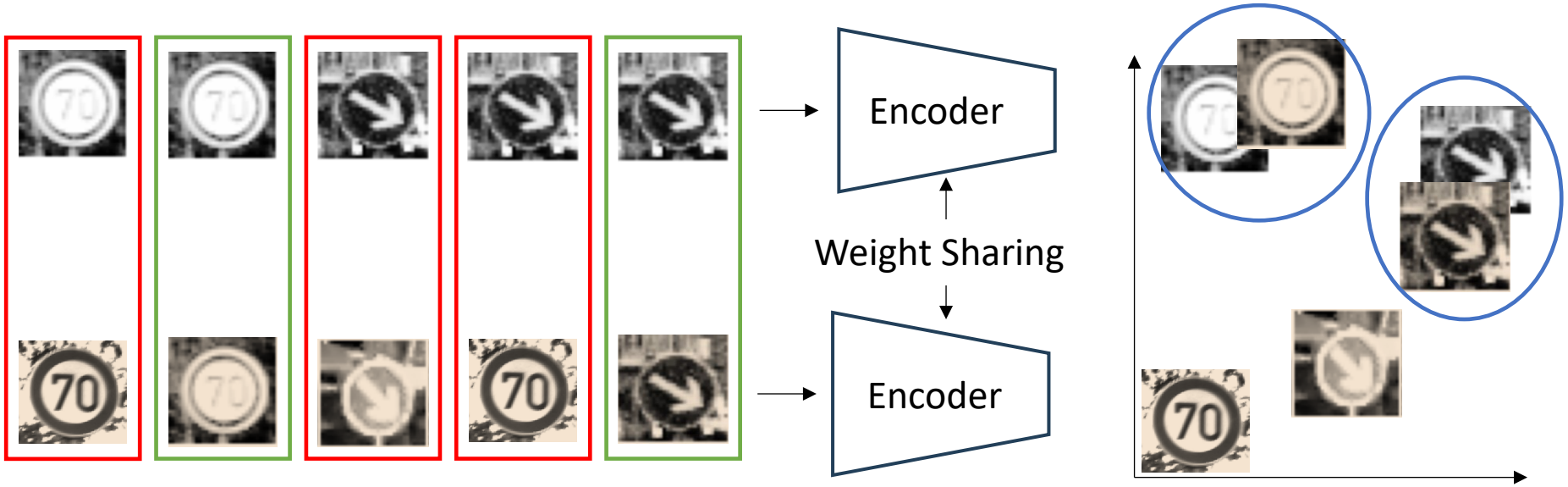
- Given positive and negative examples it is possible to learn high-quality representations.
- Labeled data supports contructing effective examples.

Contrastive Deep Clustering



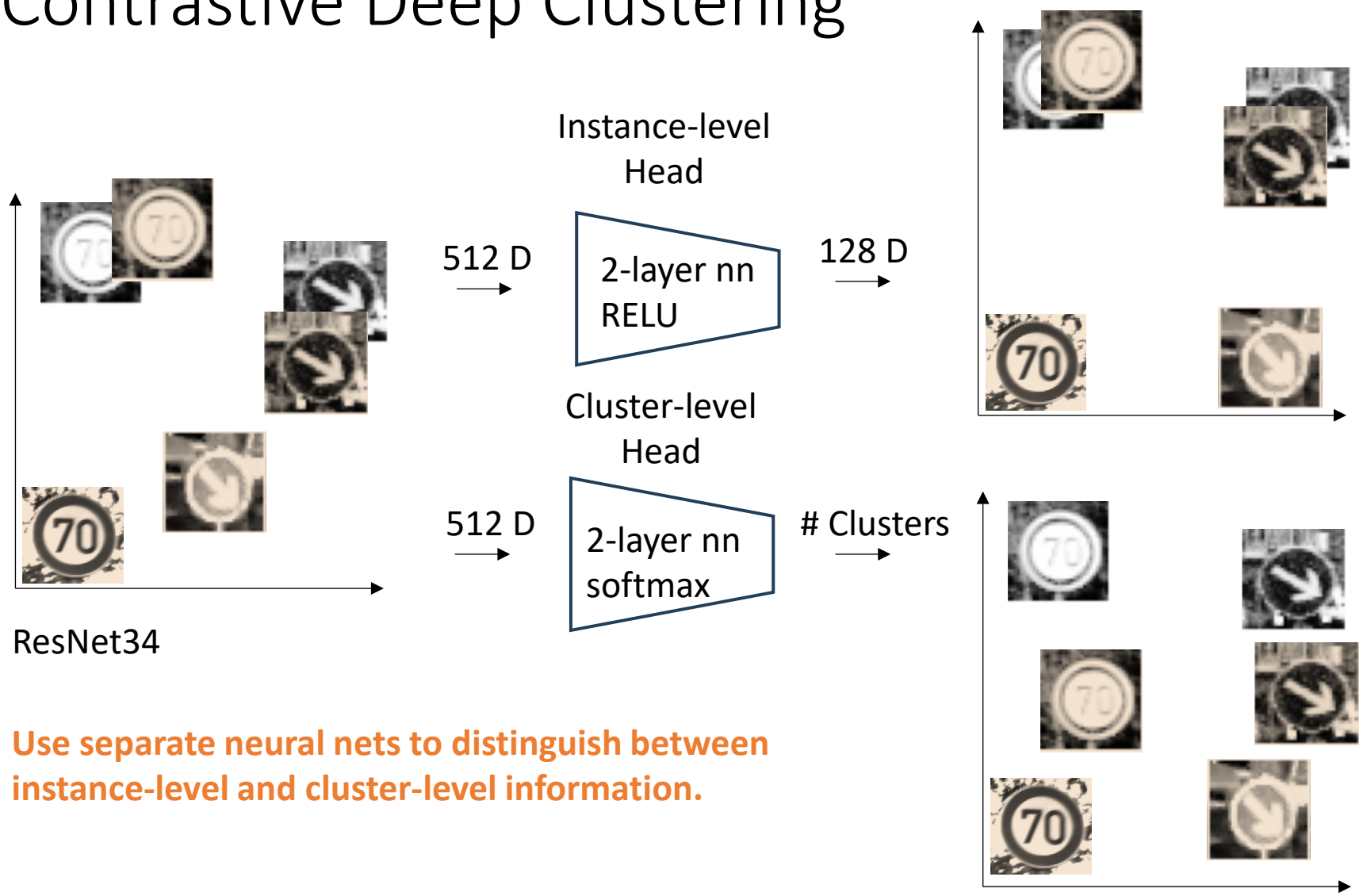
- How to construct effective positive and negative examples without labels?
- Augmentation.

Contrastive Deep Clustering



- How to construct effective positive and negative examples without labels?
- Augmentation.
- **Challenge: Finding the right type and strength of augmentation, balance between abstraction and representation.**

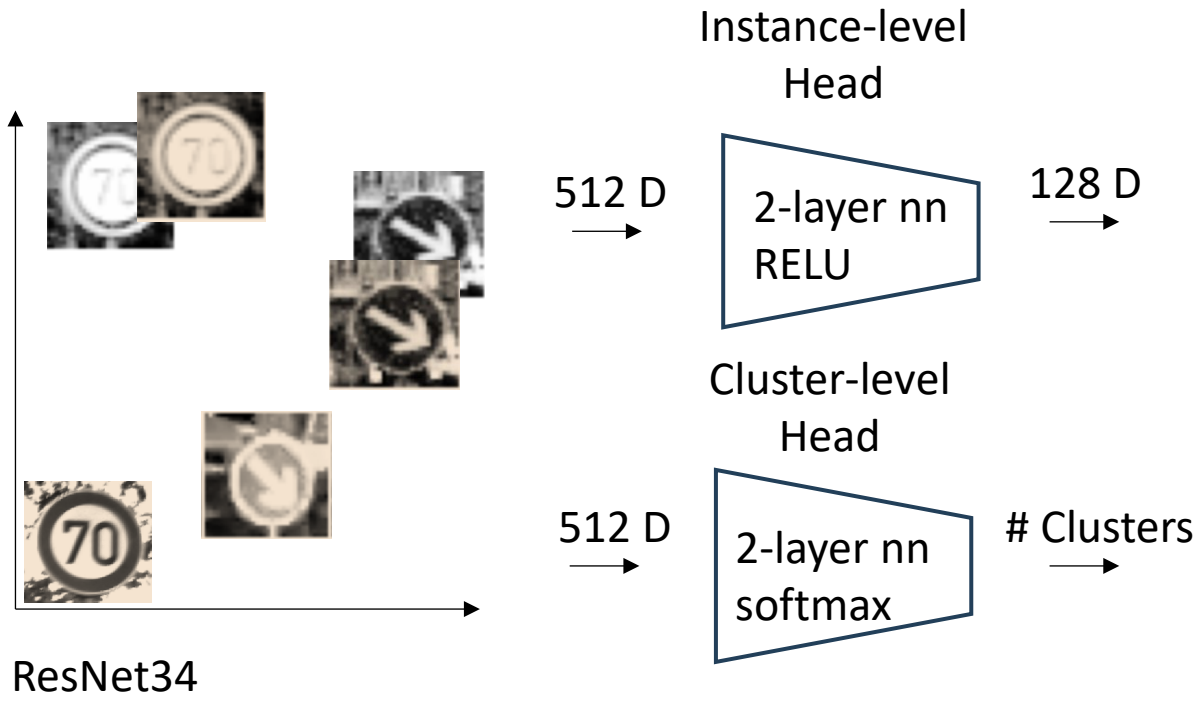
Contrastive Deep Clustering



Use separate neural nets to distinguish between instance-level and cluster-level information.

Li et al.: Contrastive Clustering. AAAI 2021

Contrastive Deep Clustering



Use separate neural nets to distinguish between instance-level and cluster-level information.

Info-NCE Loss

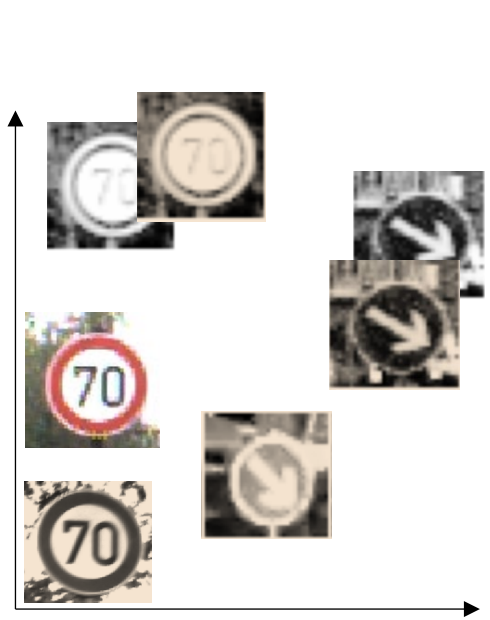
$$\mathcal{L}_{ins} = \frac{1}{2N}\sum_{i=1}^{N}(\ell_i^a + \ell_i^b)$$

consistency balance

$$\mathcal{L}_{clu} = \frac{1}{2M}\sum_{i=1}^{M}(\hat{\ell}_i^a + \hat{\ell}_i^b) - H(Y)$$

Li et al.: Contrastive Clustering. AAAI 2021

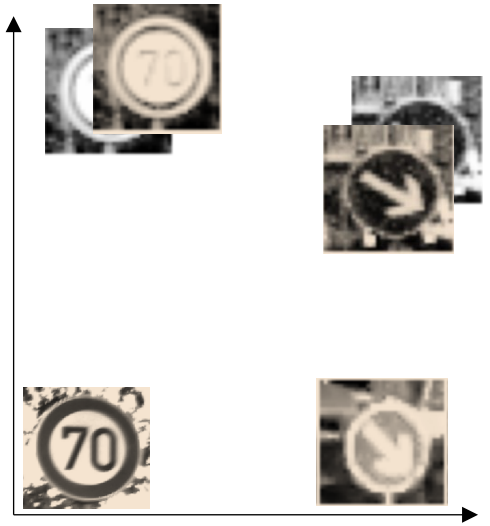
Contrastive Deep Clustering



ResNet34

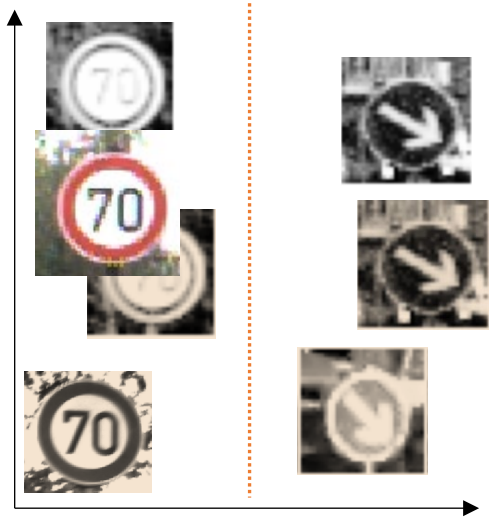
Instance-level Head

512 D → 2-layer nn RELU → 128 D



Cluster-level Head

512 D → 2-layer nn softmax → # Clusters



Use separate neural nets to distinguish between instance-level and cluster-level information.

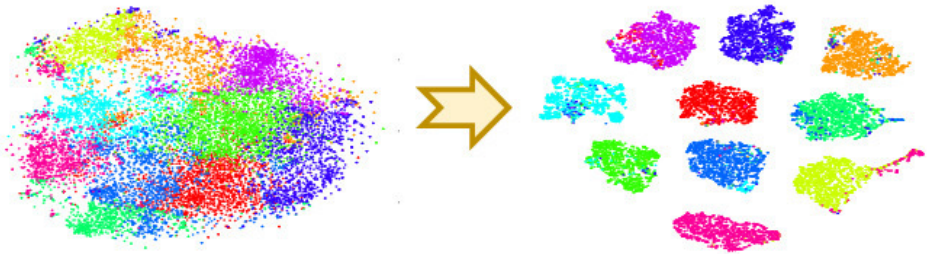
At test time:

- Extract features,
- Cluster head outputs cluster label.

Li et al.: Contrastive Clustering. AAAI 2021

Alternative Methods Building upon Contrastive Pre-training

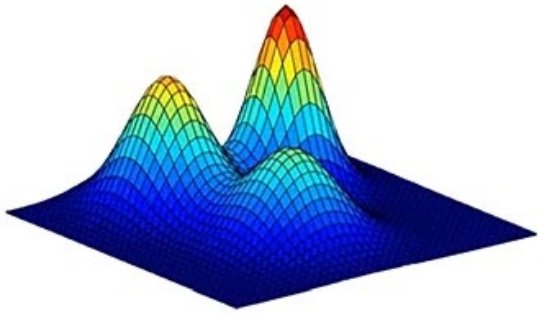
- SCAN: after pre-training, group objects and their nearest neighbors togehter, gradual cluster assignment by labeling objects with clear cluster assignment
- NNM: pairs from local, i.e. batch-wise and global nearest neighbors loss function that enforces consistency of cluster assignments
- discDC: discriminate analysis within an encoder/decoder framework to enhance cluster separation



Gansbeke et al.: SCAN: Learning to Classify Images without Labels. ECCV 2020
Dang et al. Nearest Neighbor Matching for Deep Clustering. CVPR 2021
Cai et al. discDC: Unsupervised discriminative deep image clustering via confidence-driven self-labeling. Pattern Recognition 2026

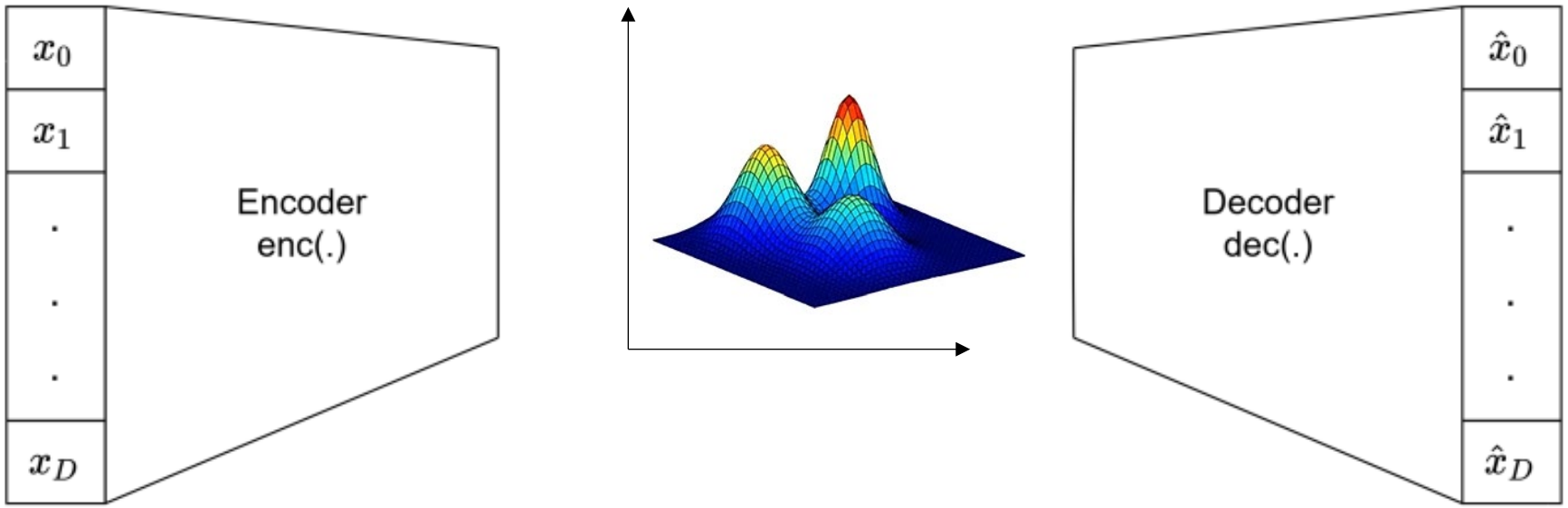
Generative Deep Clustering Methods

- Require a prior of on the data distribution of the clusters in the latent space, mostly Gaussian.
- Support generating new instances.



Generative Deep Clustering

Require a distribution prior, e.g., VaDE a mixture of Gaussians within a **Generative Autoencoder** framework.

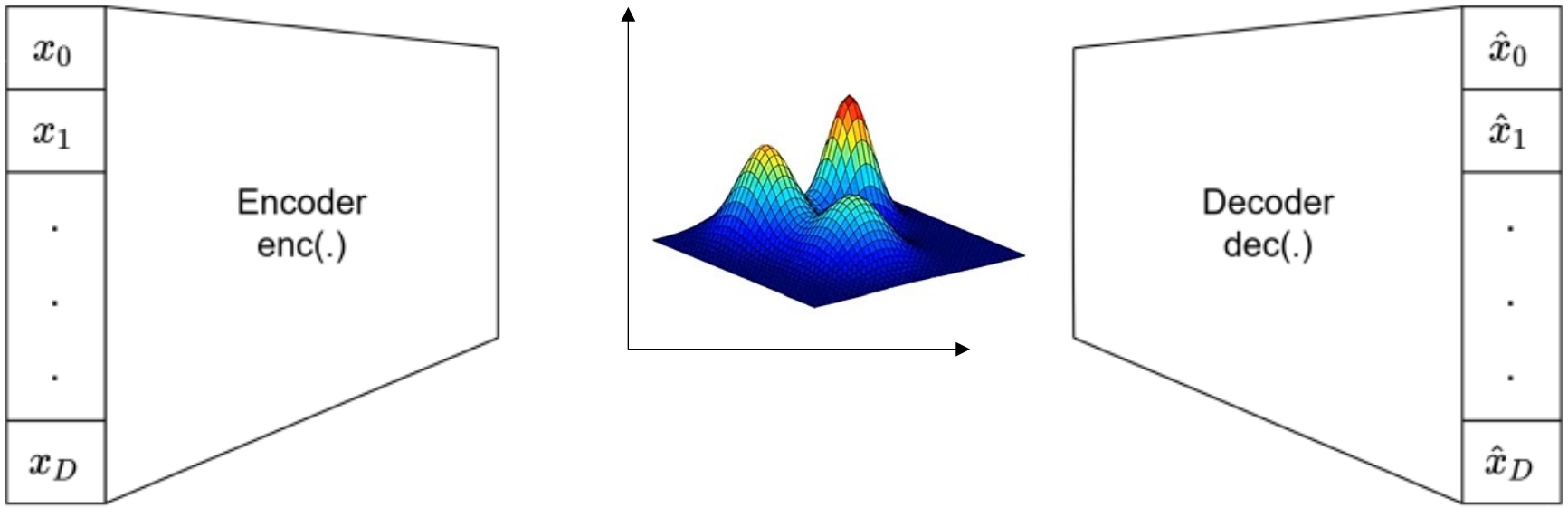


$$\mathcal{L}_{\text{ELBO}}(\mathbf{x}) = E_{q(\mathbf{z},c|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - D_{KL}(q(\mathbf{z},c|\mathbf{x})||p(\mathbf{z},c))$$

Jiang, et al.. Variational Deep Embedding: An Unsupervised and Generative Approach to Clustering. IJCAI 2017

Generative Deep Clustering

Require a distribution prior, e.g., VaDE a Gaussian Mixture Model within a **Generative Autoencoder** framework.

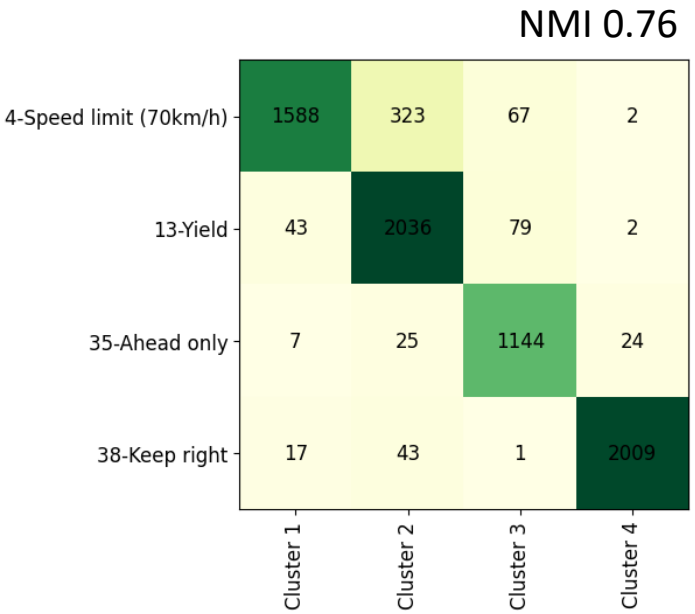


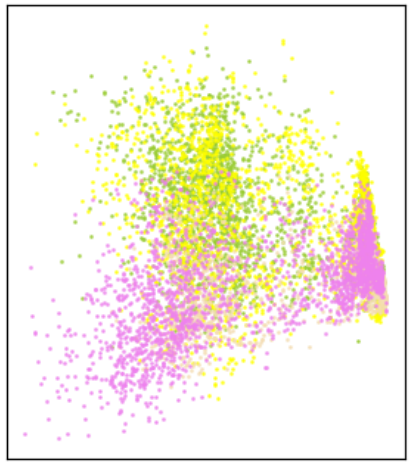
Representation Abstraction

$$\mathcal{L}_{\text{ELBO}}(\mathbf{x}) = E_{q(\mathbf{z},c|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - D_{KL}(q(\mathbf{z},c|\mathbf{x})||p(\mathbf{z},c))$$

Jiang, et al.. Variational Deep Embedding: An Unsupervised and Generative Approach to Clustering. IJCAI 2017

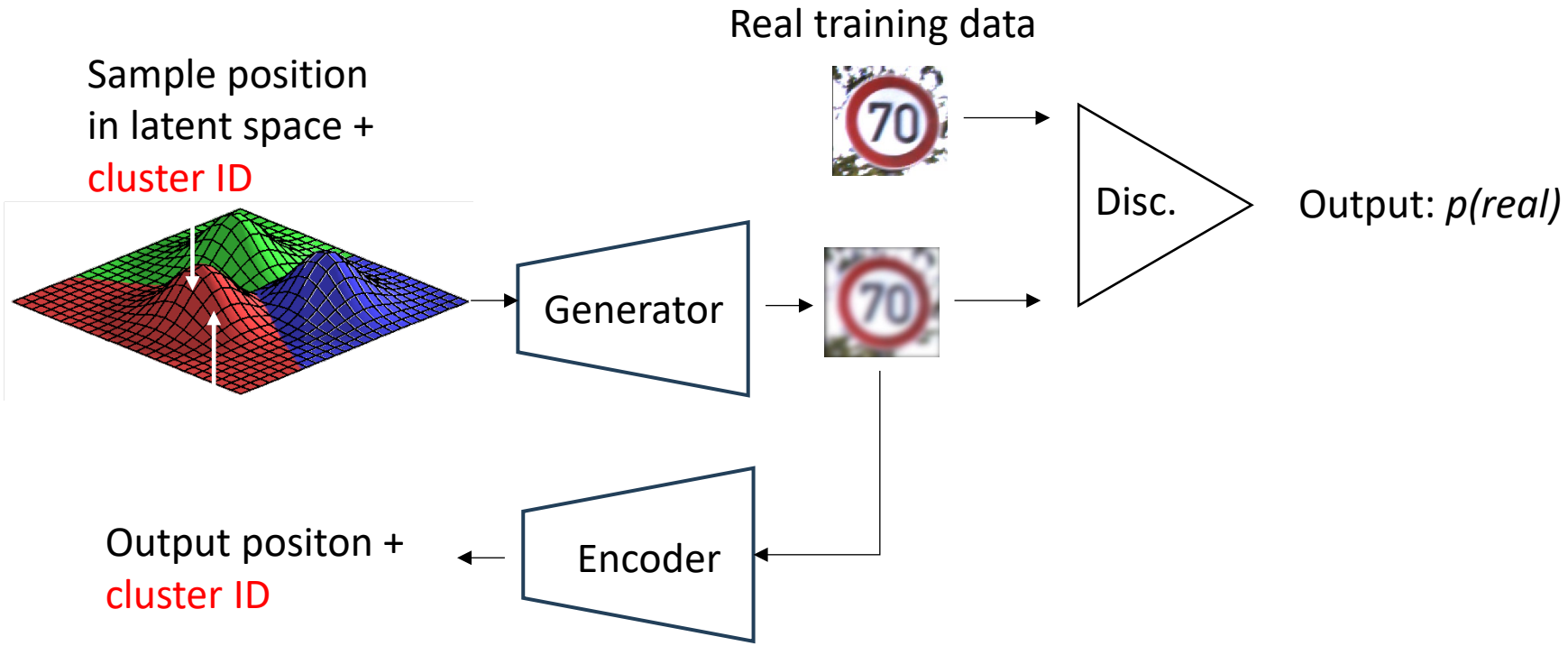
VADE on Running Example





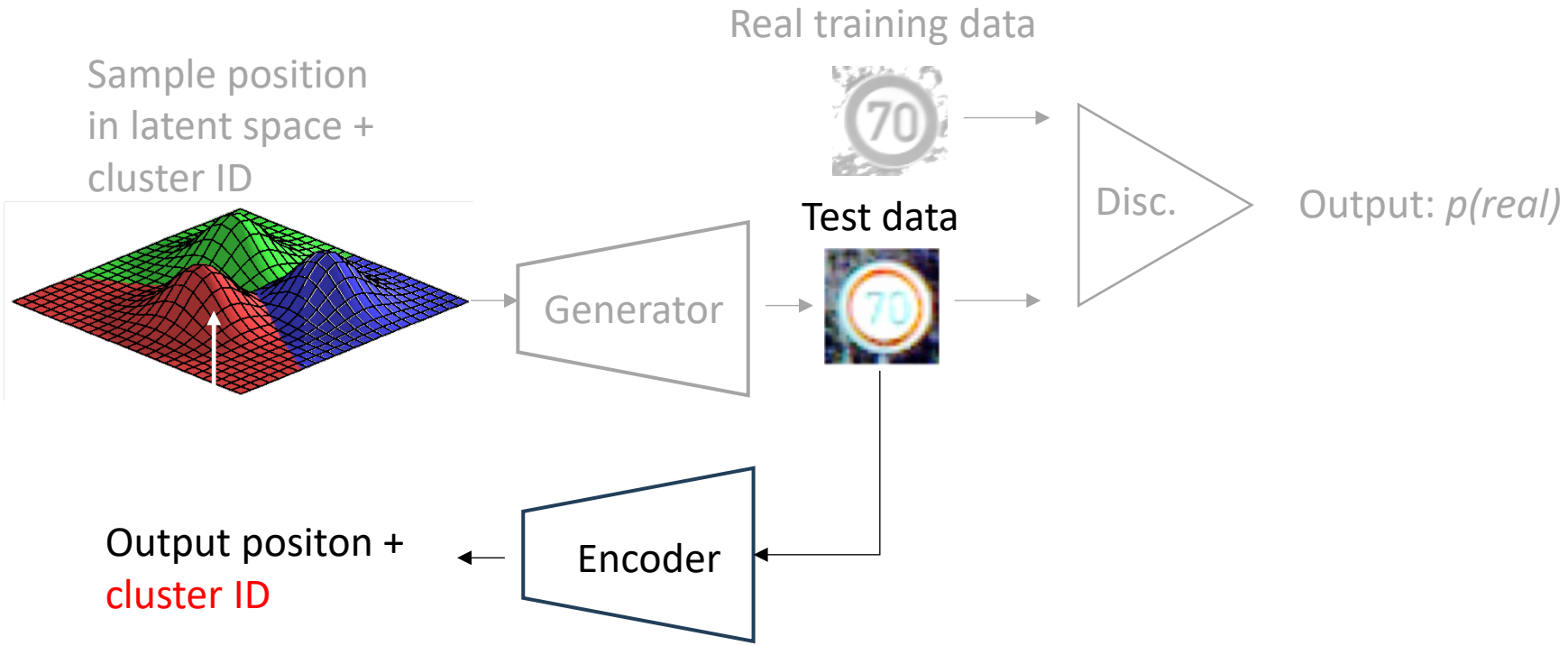
Gaussian Mixture Model prior fits the data well.

Generative Deep Clustering based on Generative Adversarial Networks



Mukherjee, et al: ClusterGAN: Latent Space Clustering in Generative Adversarial Networks. AAAI 2019

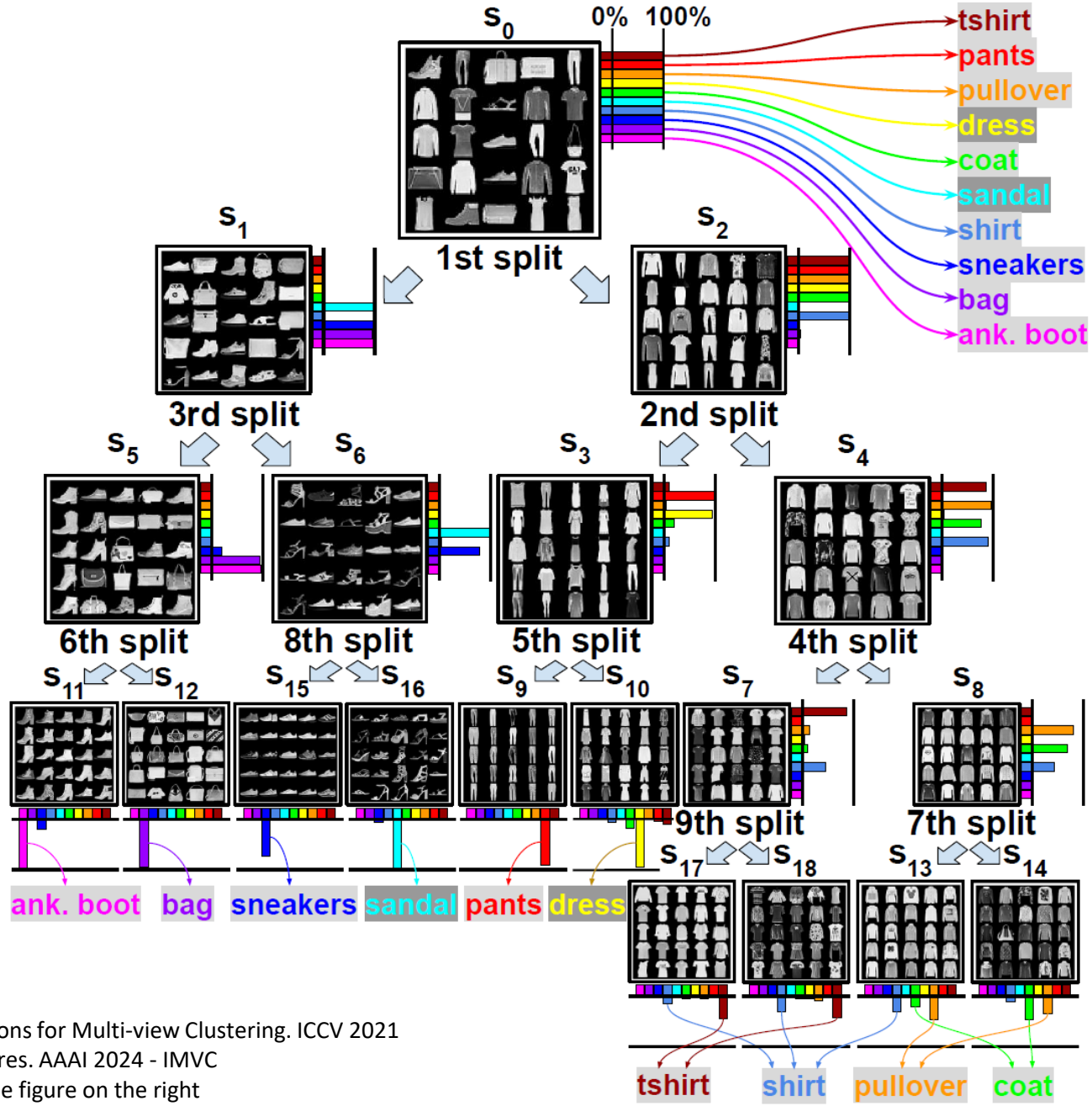
Generative Deep Clustering based on Generative Adversarial Networks: Test



Mukherjee, et al: ClusterGAN: Latent Space Clustering in Generative Adversarial Networks. AAAI 2019

Recent Methods

- Multi-VAE: VAE-based method, distinguishing between shared and different information across views
- IMVC: VAE-based method for multi-view data supporting incomplete input data
- HCMGAN: Hierarchical deep clustering method based on a GAN architecture with multiple encoders



Xu et al. Multi-VAE: Learning Disentangled View-common and View-peculiar Visual Representations for Multi-view Clustering. ICCV 2021
Xu et al. Deep Variational Incomplete Multi-View Clustering: Exploring Shared Clustering Structures. AAAI 2024 - IMVC
de Mello et al. Top-Down Deep Clustering with Multi-Generator GANs. AAAI 2022. - HCMGAN see figure on the right

Discussion of Deep Clustering Methods

Autoencoder-based:

+ well-investigated, data-efficient, not much domain knowledge needed
- contrastive methods can show stronger performance

Contrastive:

+ highly competitative
- need much training data and augmentations that require domain knowledge

Generative:

+ support generating new data
- need much training and require strong distribution assumptions

Hybrid methods might be the future

	High-dimensional data	Interpretability	Carbon Footprint	Parameterization
Classical clustering	---	+++	+++	-
Subspace clustering	+	++	++	--
Deep clustering	+++	+	---	---
Hybrid methods	+++ Expressiveness where needed?	++ Interpetable where possible?	+ Spend effort where needed?	-- Partly automatic and interactive?

We need a cost model/objective function for hybrid methods

that supports answering the fundamental question:

How to find the best trade-off between abstraction and representation:

- For a specific data set,
- For a certain cluster,
- For each individual object.

We need a cost model/objective function for hybrid methods

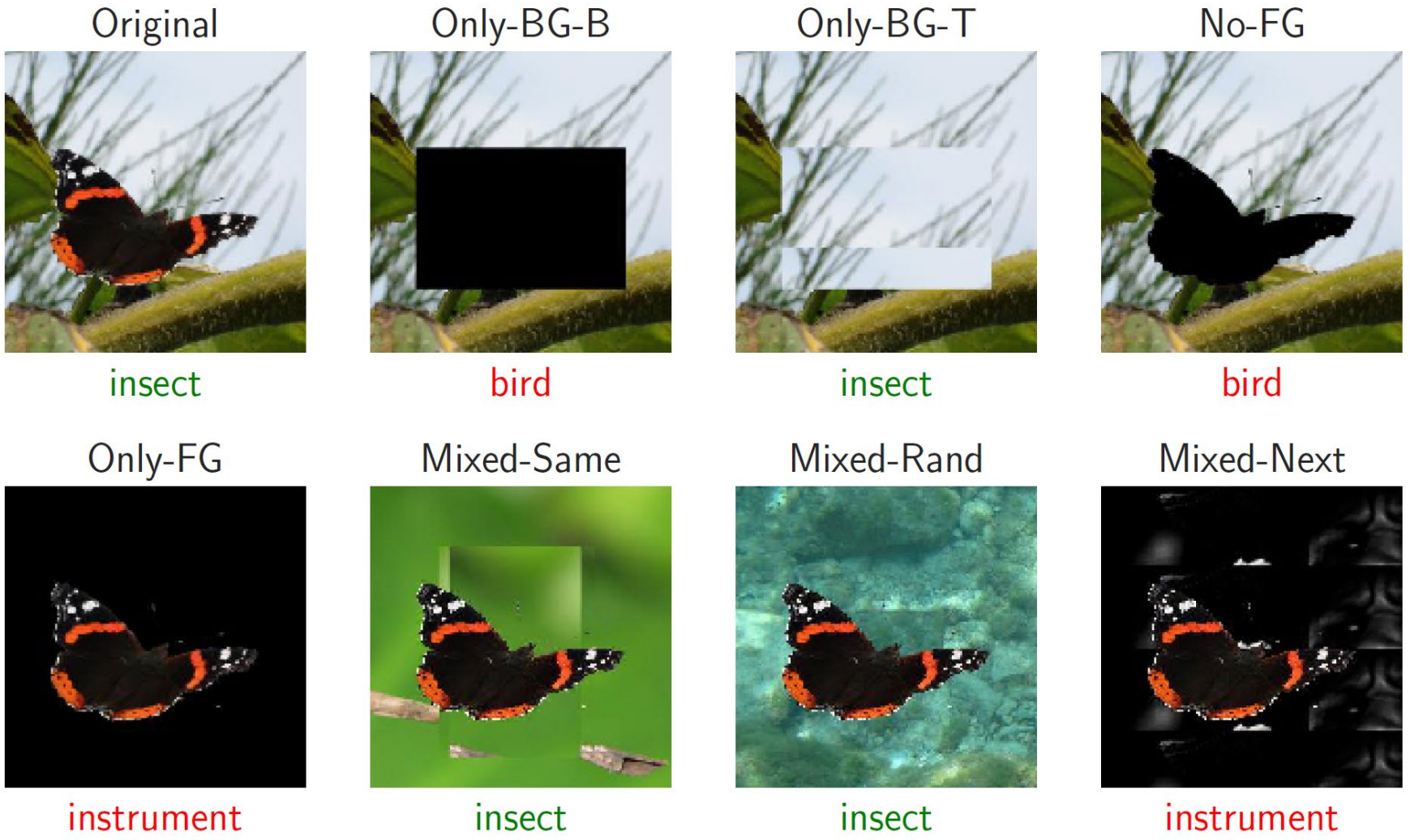
that supports answering the fundamental question:

How to find the best trade-off between abstraction and representation:

- For a specific data set,
- For a certain cluster,
- For each individual object.

Understanding this trade-off means teaching deep neural networks to cluster. Might also help for adversarial robustness, few-shot learning and regularization of supervised neural networks.

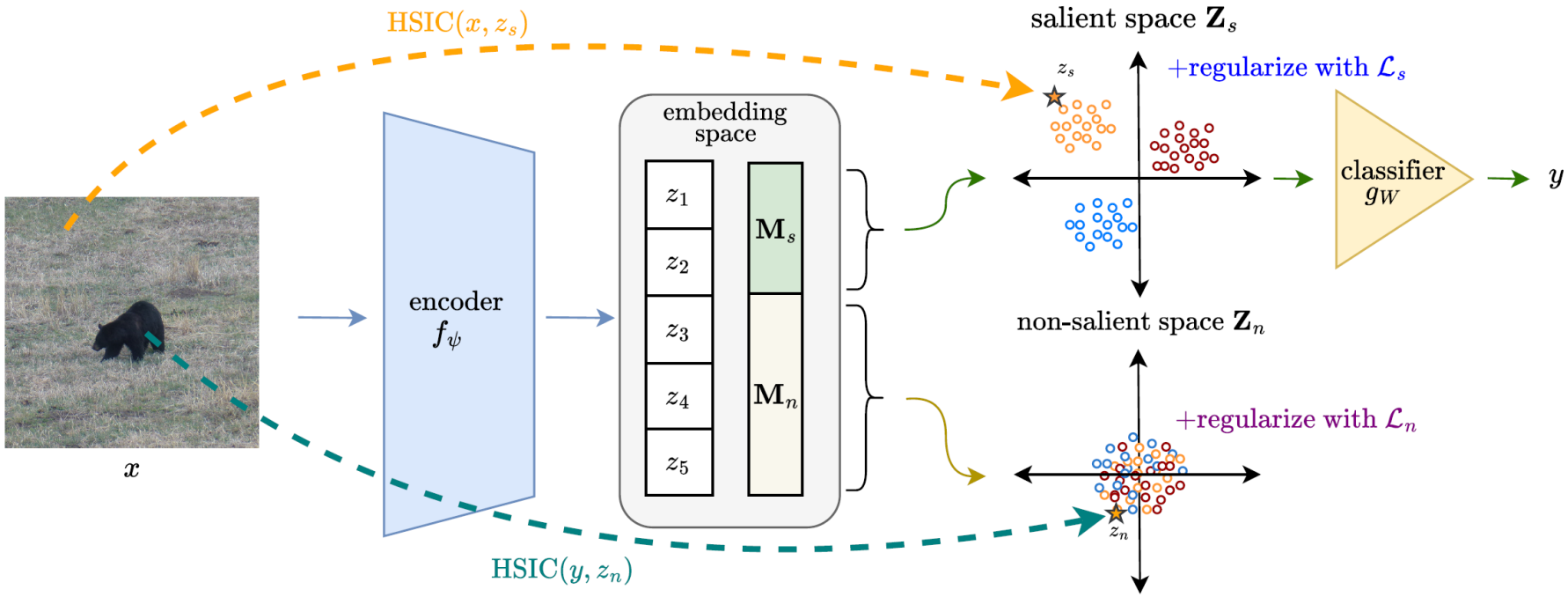
Missing Generalization in Supervised Learning



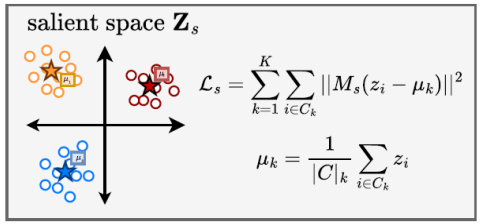
ResNet50,
25 million
parameters,
trained on over 1
million of images

Xiao et al. Noise or Signal: The Role of Background in Object Recognition, arXiv 2020

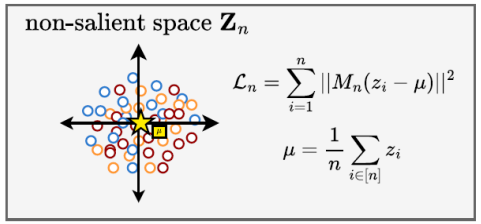
Robust Classification with Clustering Ideas



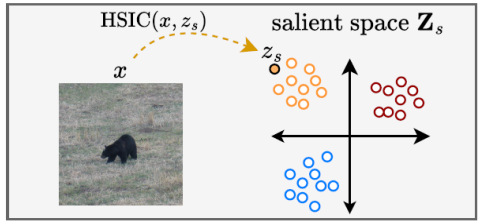
$$\mathcal{L}(\mathcal{D}; \psi, \mathbf{W}, \mathbf{M}_s, \mathbf{M}_n) = \lambda_{ce}\mathcal{L}_{ce} + \lambda_s\mathcal{L}_s + \lambda_n\mathcal{L}_n + \rho_s \text{HSIC}(\mathbf{X}, \mathbf{Z}_s) + \rho_n \text{HSIC}(\mathbf{Y}, \mathbf{Z}_n)$$



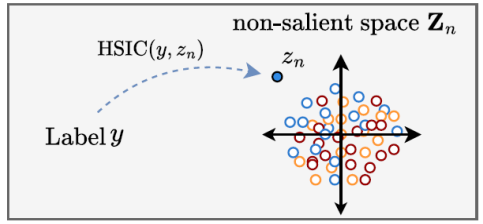
(a) $\mathcal{L}_s$
Inter-class separability



(b) $\mathcal{L}_n$
Global invariance



(c) HSIC(X, Z_s)
Compress redundant information



(d) HSIC(Y, Z_n)
Decorrelate label information

L. Miklautz et al. H-SPLID: HSIC-based Saliency Preserving Latent Information Decomposition. NeurIPS 2025.

Want to contribute?

Tutorial Website