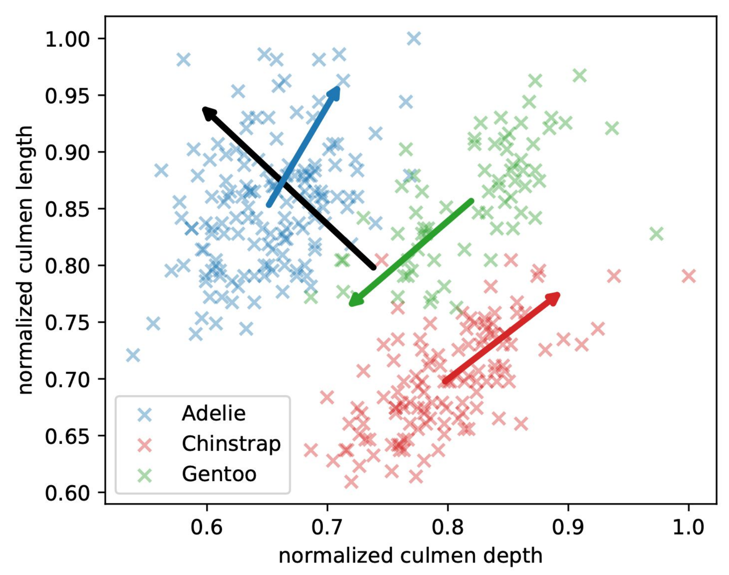
# Improved Representation Learning Through Tensorized Autoencoders

Pascal Esser[*], Satyaki Mukherjee[*], Mahalakshmi Sabanayagam[*], Debarghya Ghoshdastidar

## Problem Setting

- Standard Autoencoder (AE) learns one representation of the data.
- However this might not capture cluster structures well.

Example: Linear AE Learns principal components of data. However this might not capture clusters: *first principal component of clusters are plotted in red, blue and green. Principal direction for the full dataset in black.*

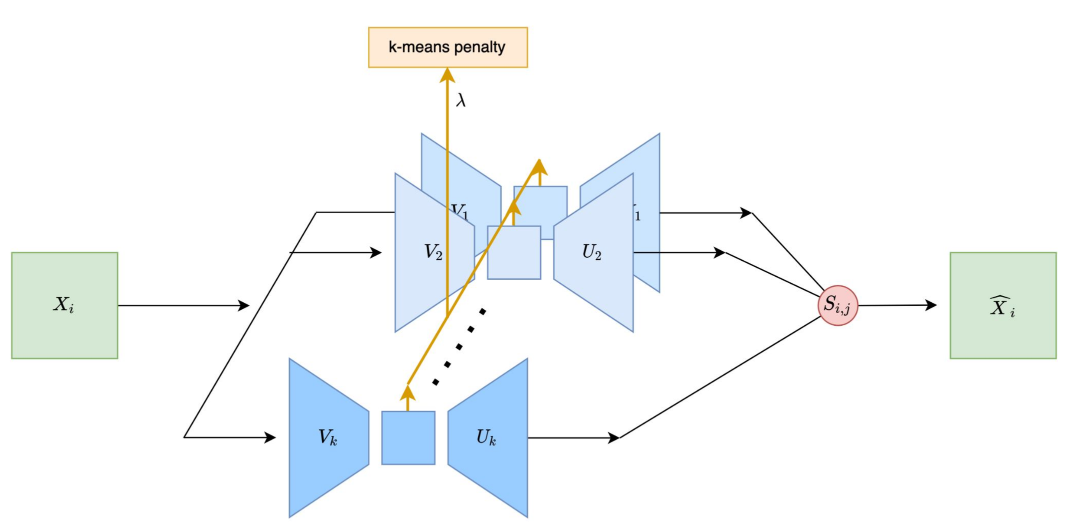



## Our Approach: Tensorized Autoencoder (TAE)

Main Idea:

We want to

- Learn cluster assignment and embedding jointly.
- Learn one representation for each cluster (one AE per cluster).




Formally we minimize

$$\min_{\{\Phi_j,\Psi_j\}_{j=1}^k, S} \sum_{i=1}^n \sum_{j=1}^k S_{j,i}\left[\left\|(X_i - C_j) - f_{\Phi_j}\left(g_{\Psi_j}(X_i - C_j)\right)\right\|^2 - \lambda * \left\|g_{\Psi_j}(X_i - C_j)\right\|^2\right],$$

- $g_j(\,)$ is the encoder and $f_j(\,)$ is the decoder for cluster j.
- $C_j$ is the center of class j.
- $S_{ij}$ assigns a datapoint $X_i$ to j'th AE.

## Meta Algorithm

1. Initialize weights and cluster assignments according to $k$-means ++.
2. Update the weights for the encoder and decoder (using e.g. a GD step).
3. Update the class assignment $S$. For example using a gradient descent step under constraints.

## Parameterization at Optimum

Consider linear encoder & decoder such that the loss becomes

$$\sum_{i=1}^n \sum_{j=1}^k S_{j,i}\left[\|(X_i - C_j) - V_j U_j (X_i - C_j)\|^2 - \lambda \|U_j (X_i - C_j)\|^2\right],$$
$$\text{s.t. } \mathbf{1}_k^T S = \mathbf{1}_n^T,\ S_{j,i} \geq 0.$$

Then for $0 < \lambda \leq 1$, optimizing the above results in the parameters at the optimum satisfying the following:

1. **Class Assignment.** While in the above equation we define $S_{j,i}$ as the probability that $X_i$ belongs to class $j$ at the optimal $S_{j,i}$ = 1 or 0 and therefore converges to a strict class assignment.
2. **Centers.** $C_j$ at optimum naturally satisfies the condition

$$C_j = \frac{\sum_{i=1} S_{j,i} X_i}{\sum_{i=1} S_{j,i}}$$

3. **Encoding / Decoding (learned weights) for j'th cluster**. Encoding corresponds to top h eigenvectors of matrix

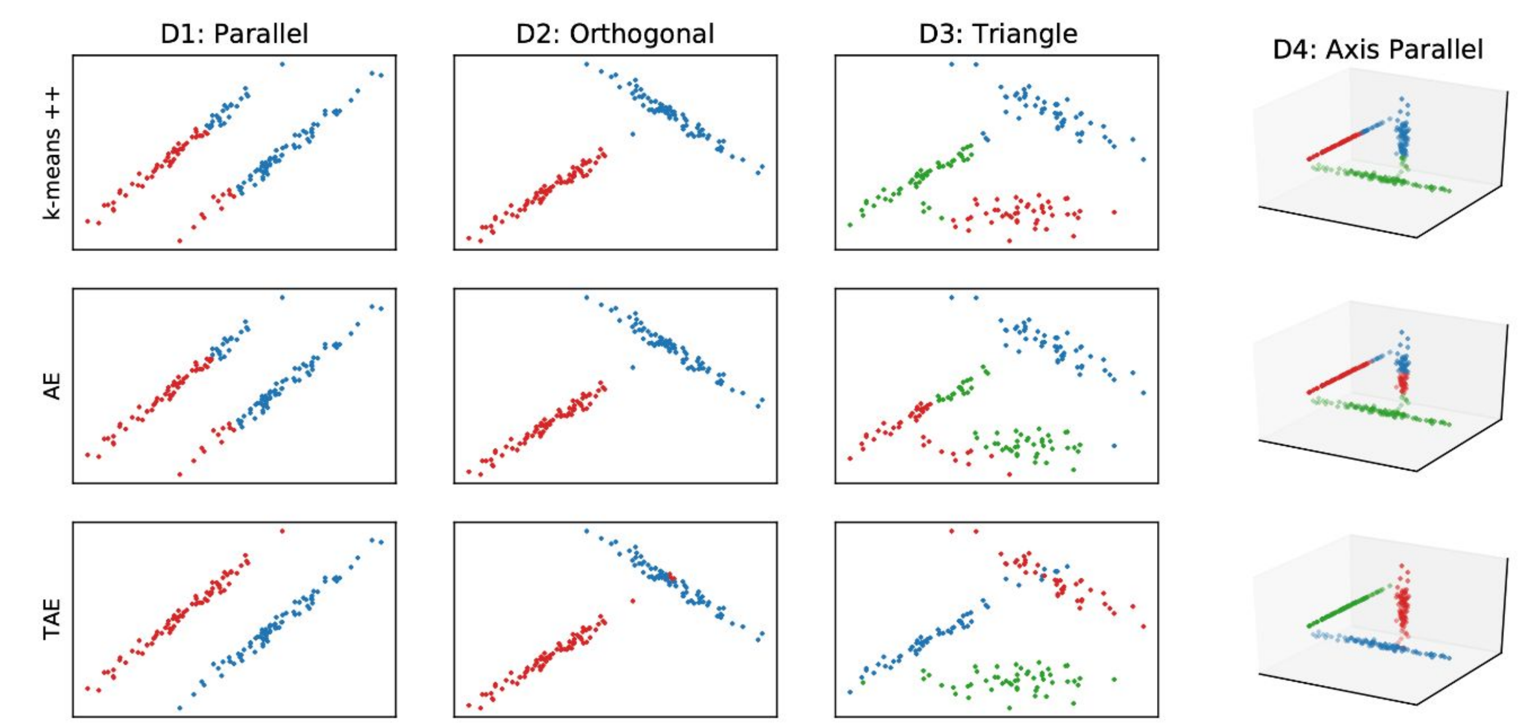
$$\hat{\Sigma}_j := \sum_{i=1}^n S_{j,i}(X_i - C_j)(X_i - C_j)^T$$

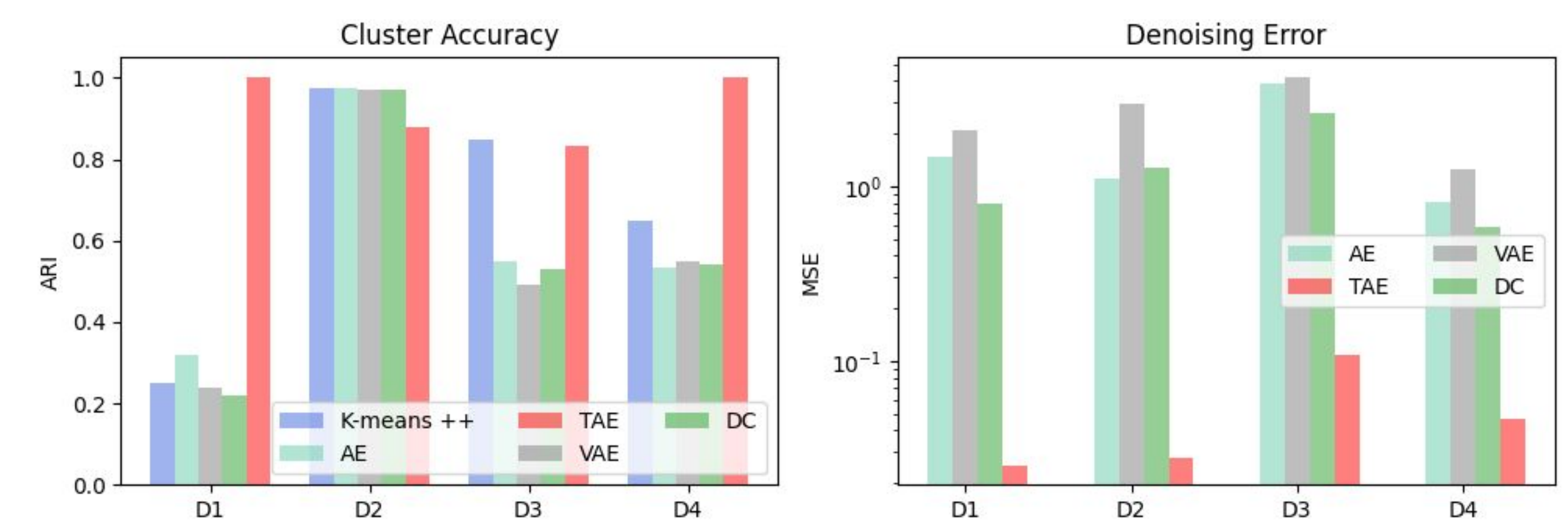
## Paper and Code

## Experimental Results

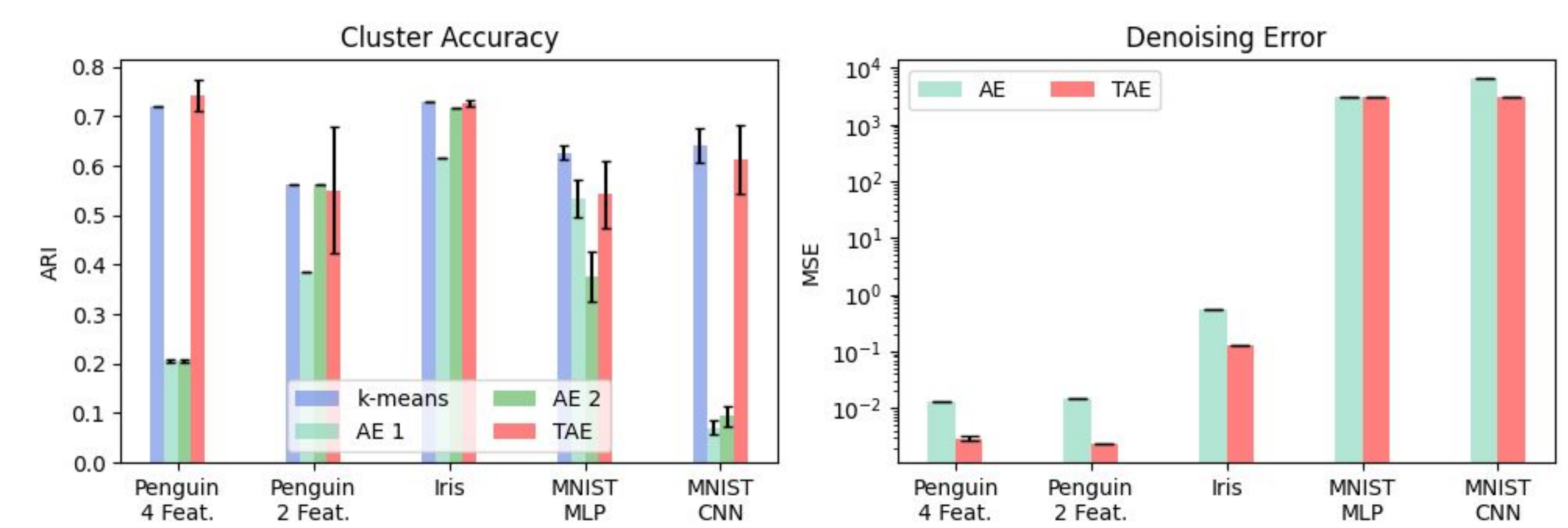
Illustration on Toy Data for Clustering and Denoising

- 5 dimensional data where three are noise dimensions




Real Data




In the paper we furthermore show:

- Additional real data experiments
- Connection to Expectation Maximization