Innovative R&D by NTT

NIPS2016

Multi-view Anomaly Detection via Robust Probabilistic Latent Variable Models

Tomoharu Iwata NTT Communication Science Labs

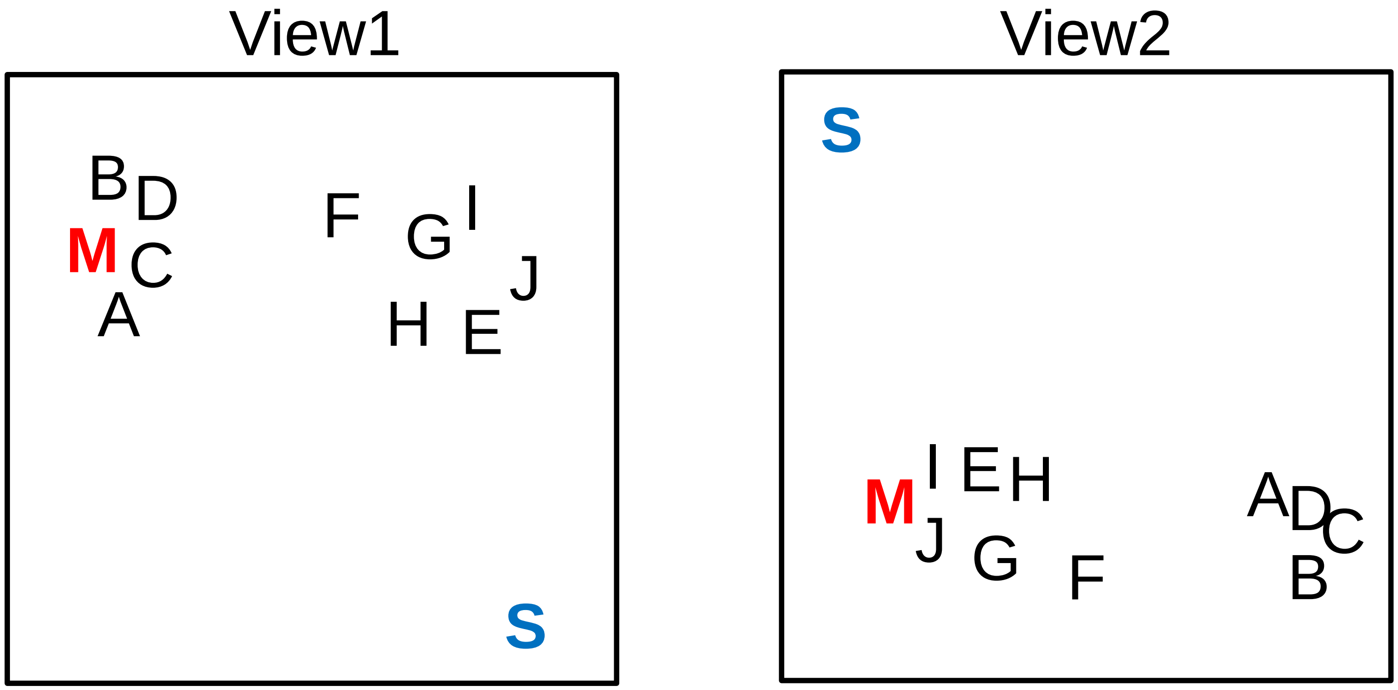
Makoto Yamada RIKEN, AIP

Multi-view anomaly

- Instances that have inconsistent views
- Application
  - information disparity management
    - find documents that contain different information across multilingual Wikipedia documents
  - malicious insider detection
  - purchase behavior analysis
    - find movies inconsistently purchased by users based on the genre (animation by purchased by grown-ups)

Multi-view data	image annotation	text URLs	audio video	Japanese text English text

Single-view/multi-view anomaly

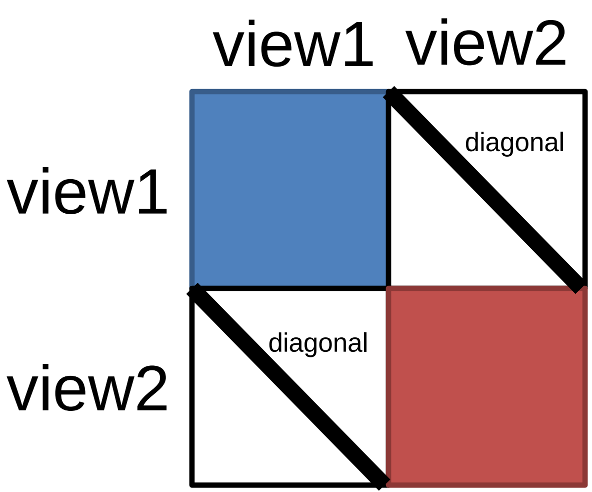


- single-view anomaly is an instance that does not conform to expected behavior
- **S** is single-view anomaly, but not multi-view anomaly
- **M** is not single-view anomaly, but multi-view anomaly

Existing method: HOAD (1)

- HOrizontal Anomaly Detection (HOAD)
  – A Spectral Framework for Detecting Inconsistency across Multi-source Object Relationships, Gao et. al. ICDM 2011
- Step1: soft clustering two views together with the constraint that an instance should be assigned to the same cluster

two-view combined similarity graph



spectral clustering

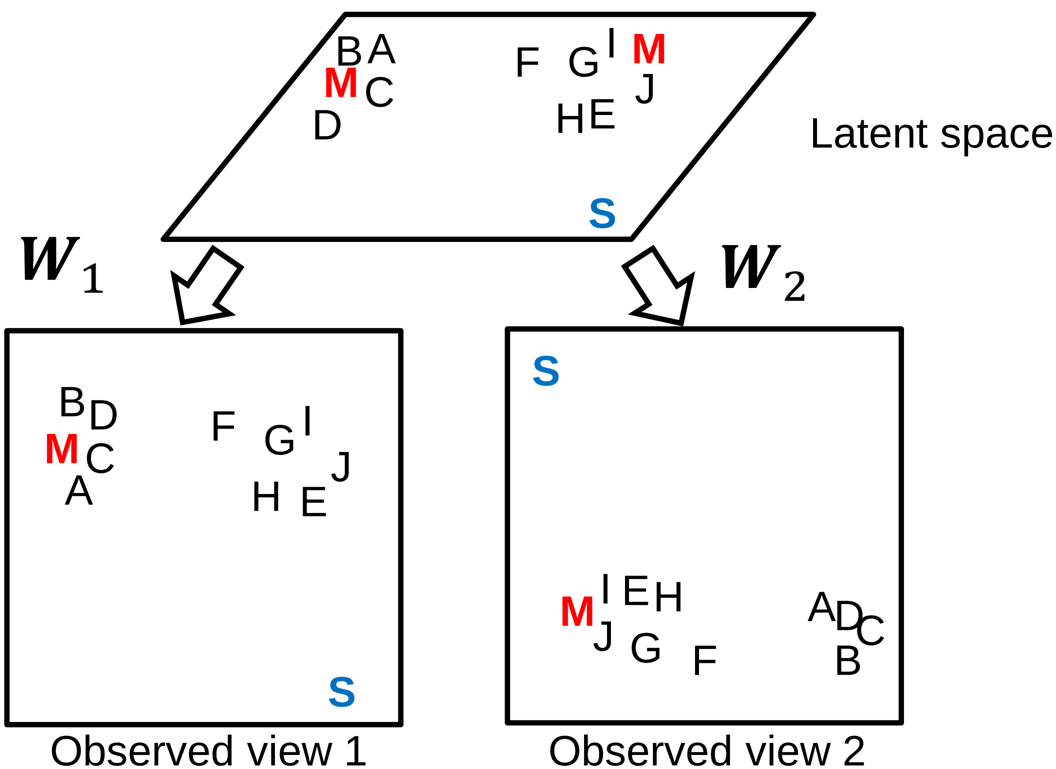
Existing method: HOAD (2)

- Step2: quantify the difference between the two clustering solutions

- weak points
  - anomalous instances also have the constraint to be assigned to the same cluster
  - require hyper-parameter tuning (e.g. weight for the constraint)

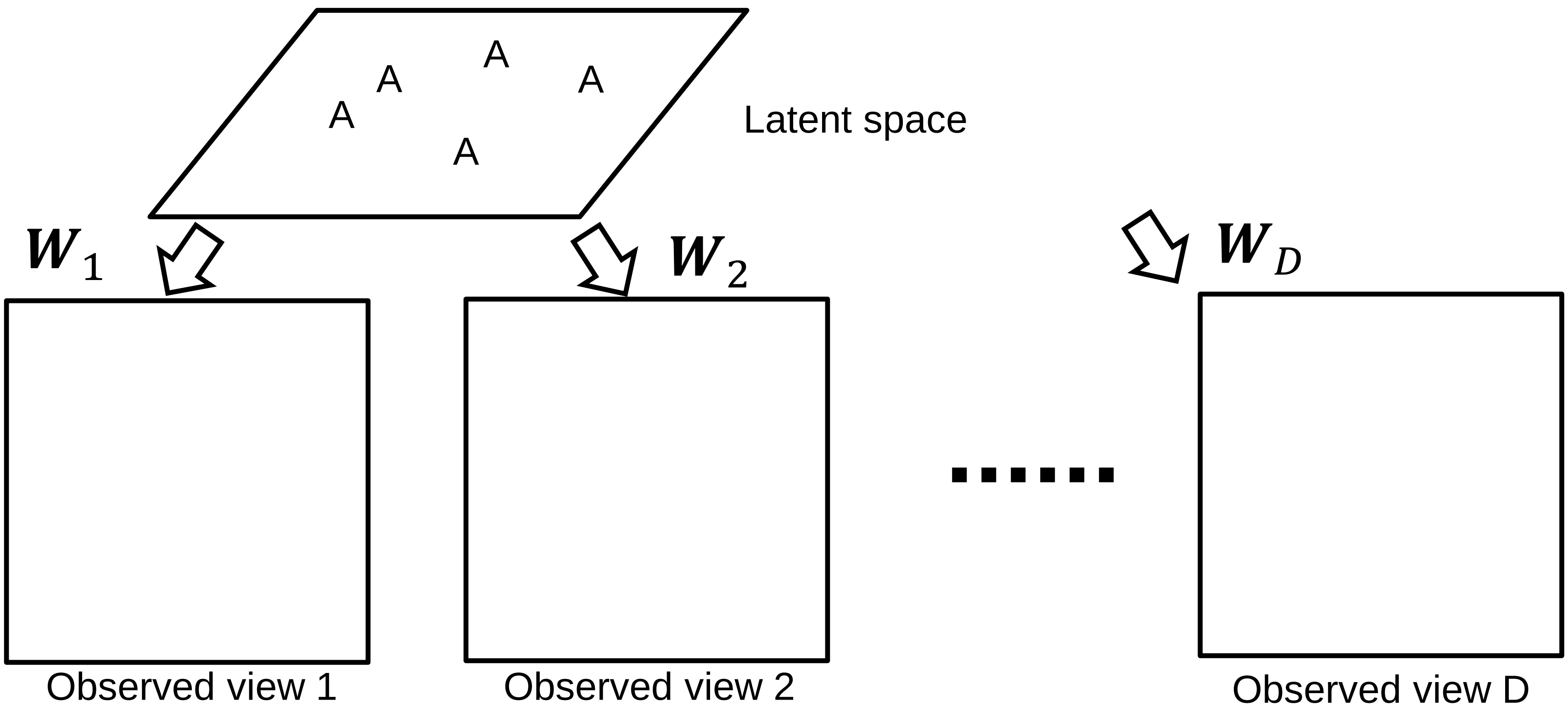
Proposed model

- Normal (non-anomalous) instance
  - all views are generated from a single latent vector
- Anomaly
  - different views are generated from different latent vectors



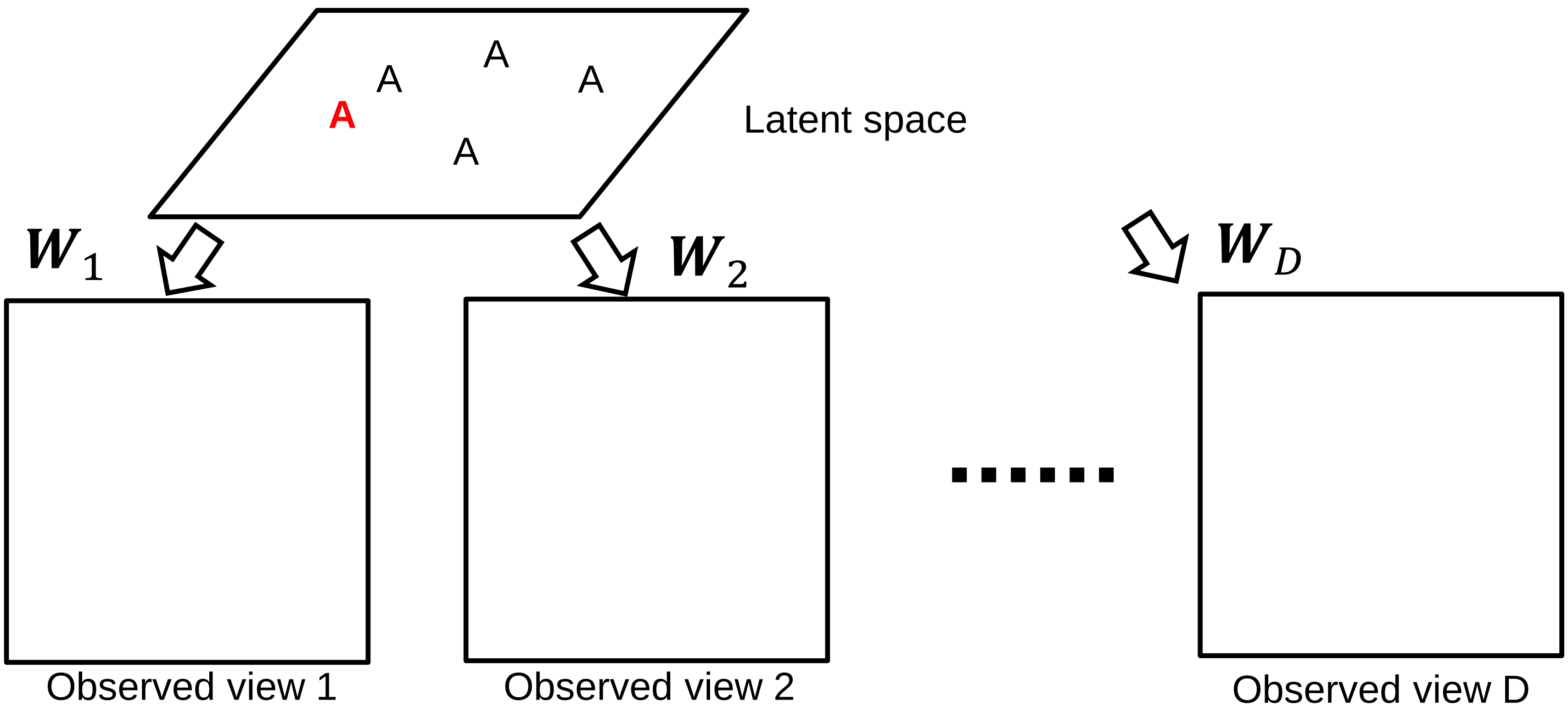
Proposed model

- Each instance has potentially a countably infinite number of latent vectors
- Each view of an instance is generated depending on a view-specific projection matrix and a latent vector



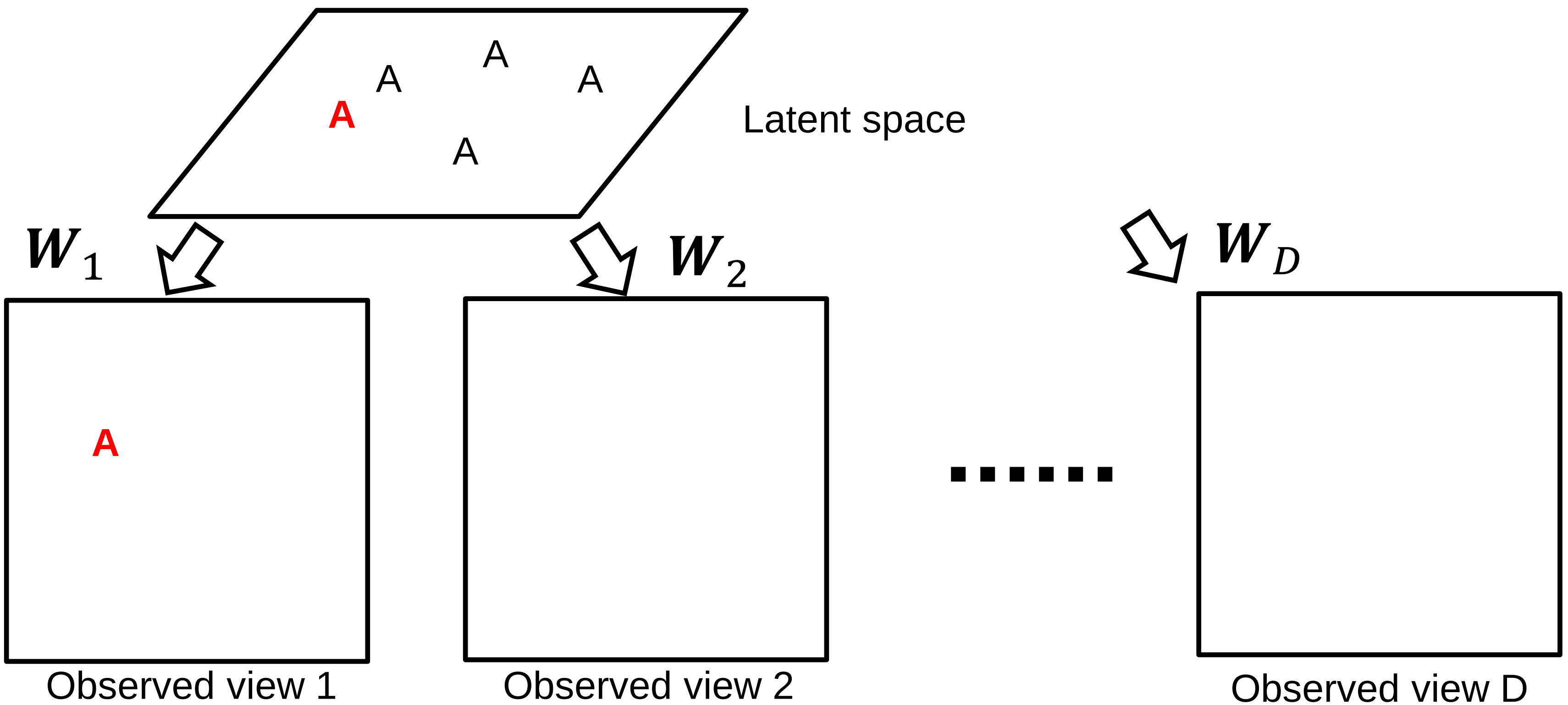
Proposed model

- Each instance has potentially a countably infinite number of latent vectors
- Each view of an instance is generated depending on a view-specific projection matrix and a latent vector



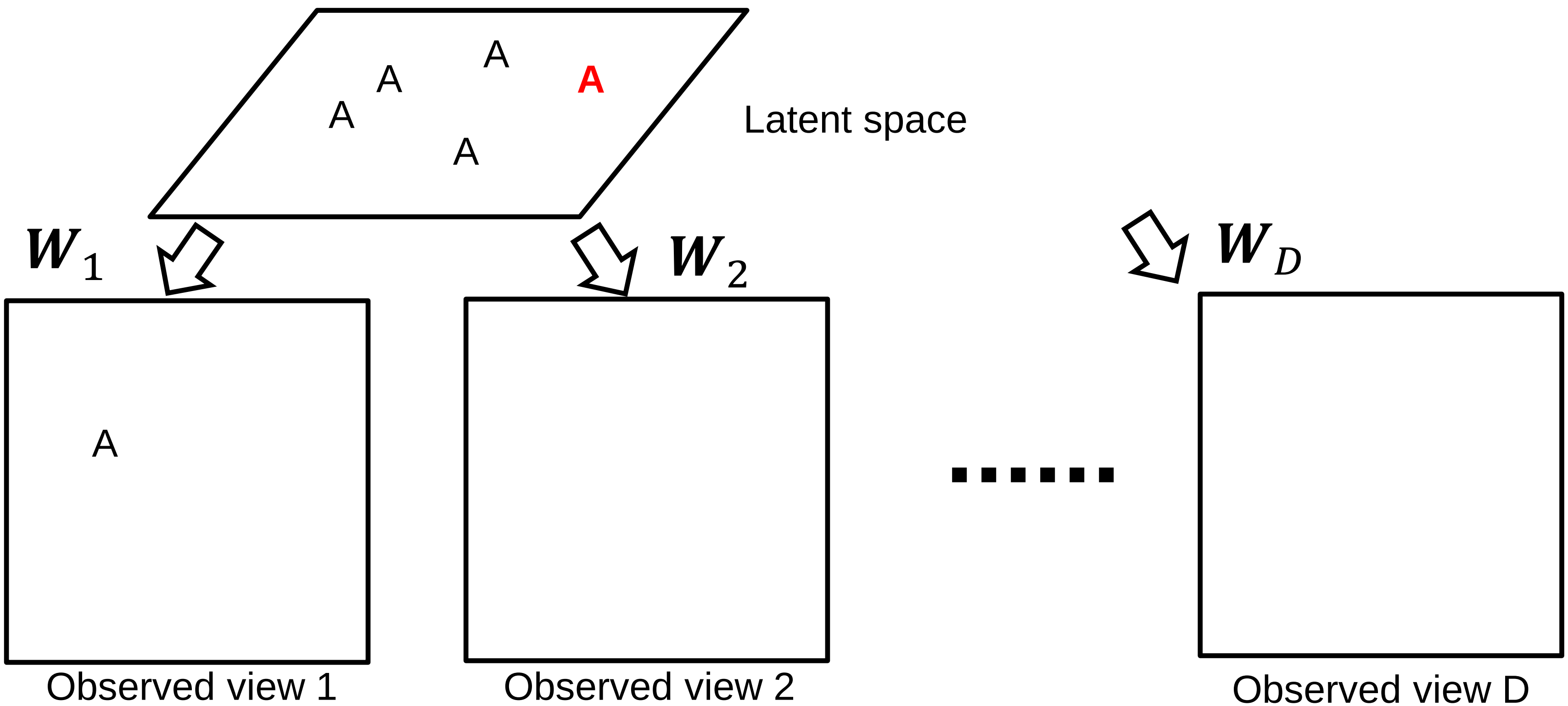
Proposed model

- Each instance has potentially a countably infinite number of latent vectors
- Each view of an instance is generated depending on a view-specific projection matrix and a latent vector



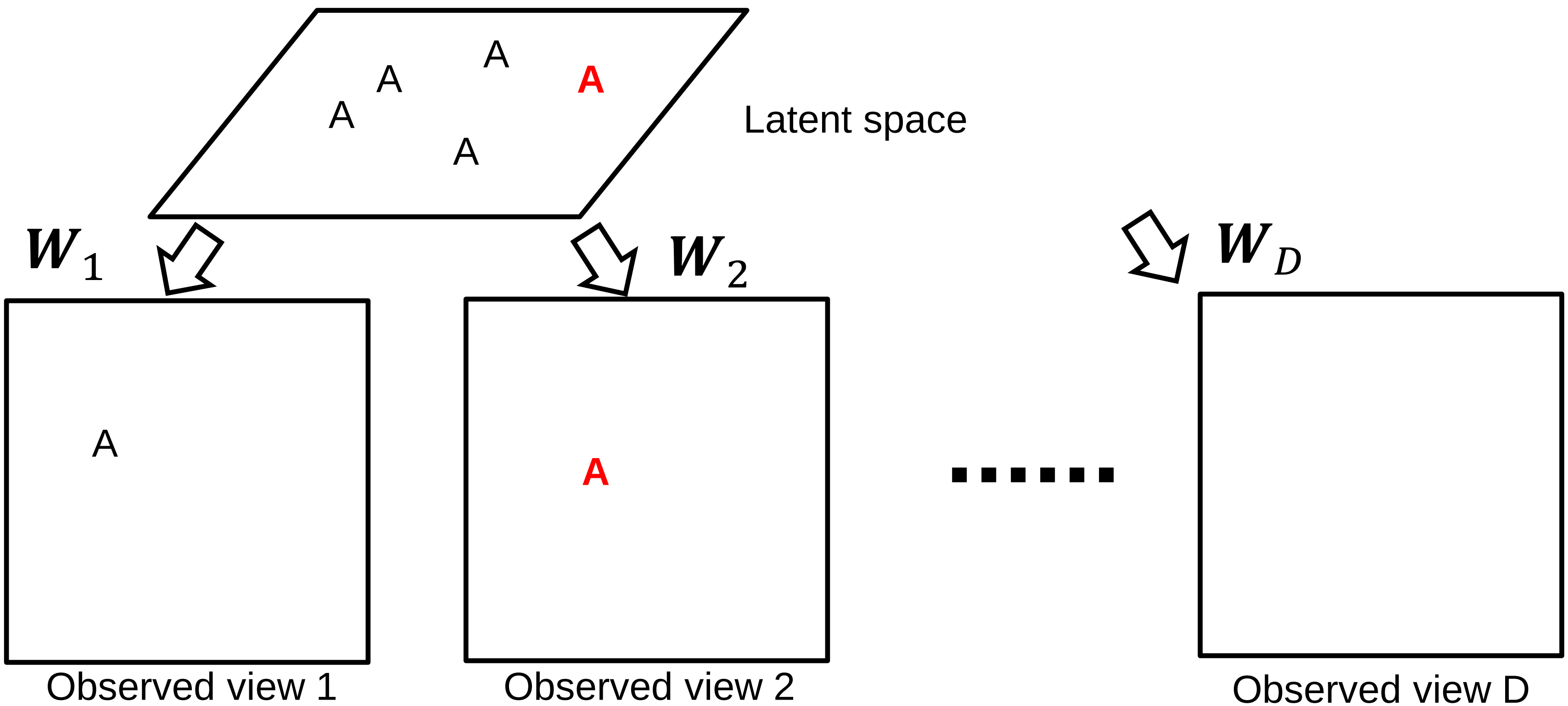
Proposed model

- Each instance has potentially a countably infinite number of latent vectors
- Each view of an instance is generated depending on a view-specific projection matrix and a latent vector



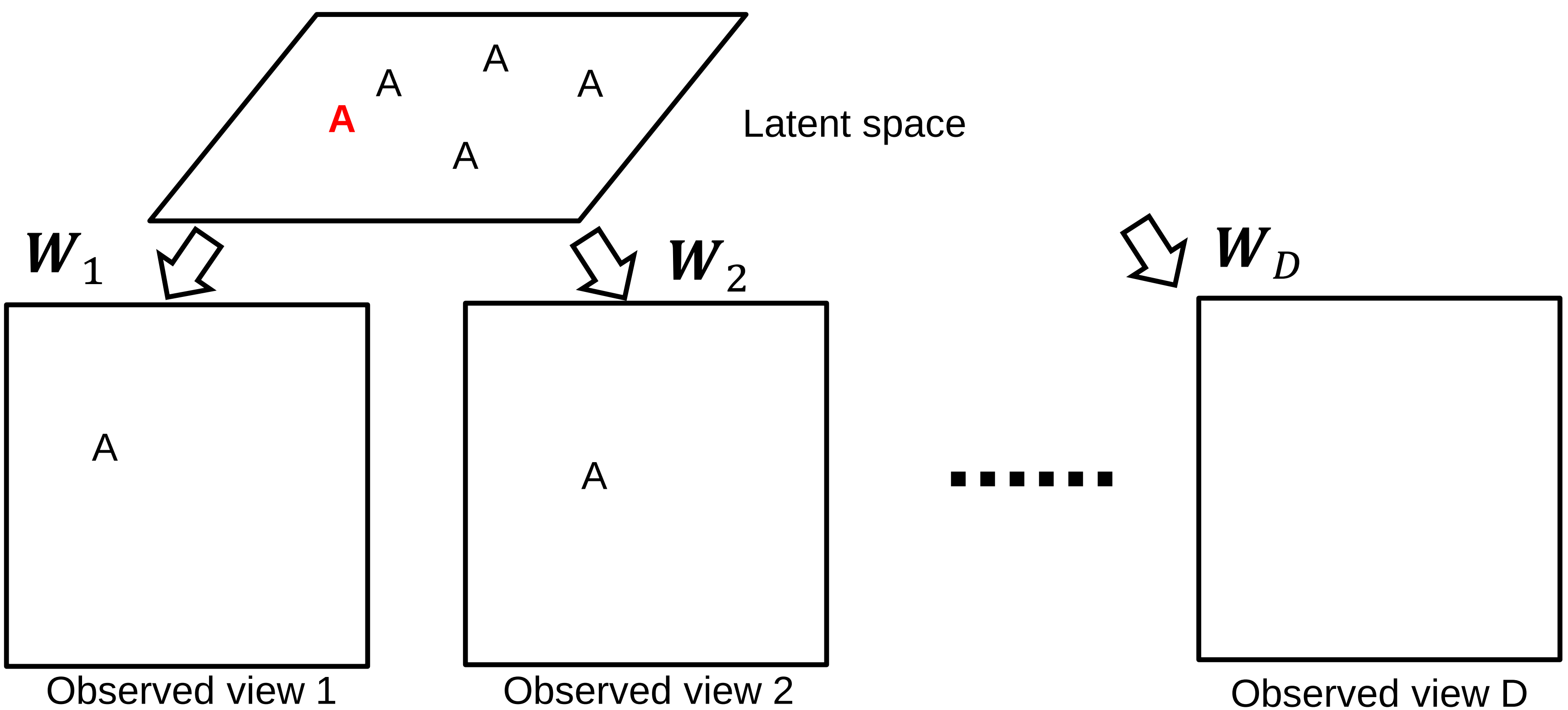
Proposed model

- Each instance has potentially a countably infinite number of latent vectors
- Each view of an instance is generated depending on a view-specific projection matrix and a latent vector



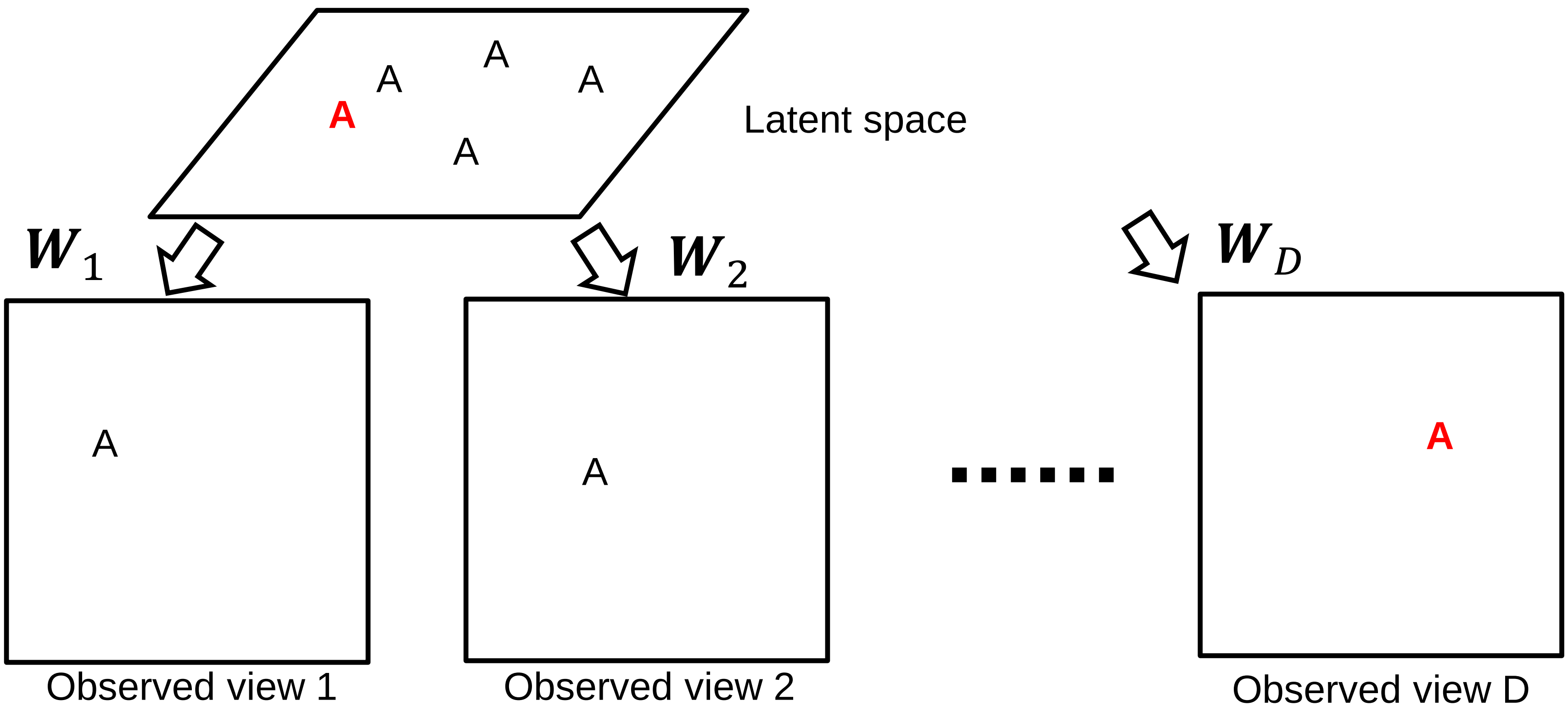
Proposed model

- Each instance has potentially a countably infinite number of latent vectors
- Each view of an instance is generated depending on a view-specific projection matrix and a latent vector



Proposed model

- Each instance has potentially a countably infinite number of latent vectors
- Each view of an instance is generated depending on a view-specific projection matrix and a latent vector



Proposed probabilistic model

$$p(\boldsymbol{x}_{nd}|\boldsymbol{z}_n, \boldsymbol{W}_d, \boldsymbol{\theta}_n, \alpha) = \sum_{j=1}^{\infty} \theta_{nj} N(\boldsymbol{x}_{nd}|\boldsymbol{W}_d \boldsymbol{z}_{nj}, \alpha^{-1}\boldsymbol{I})$$

view d of instance n

mixture weight for j-th latent vector

projection matrix for view d

precision

j-th latent vectors of instance n

- By using Dirichlet processes prior for each instance, the number of latent vectors is automatically determined from the given data
  - if views are consistent, they are clustered; otherwise, they need different latent vectors
- By using view-dependent projection matrices, we can handle different properties across different views

Anomaly score:
probability that the instance uses more than one latent vector

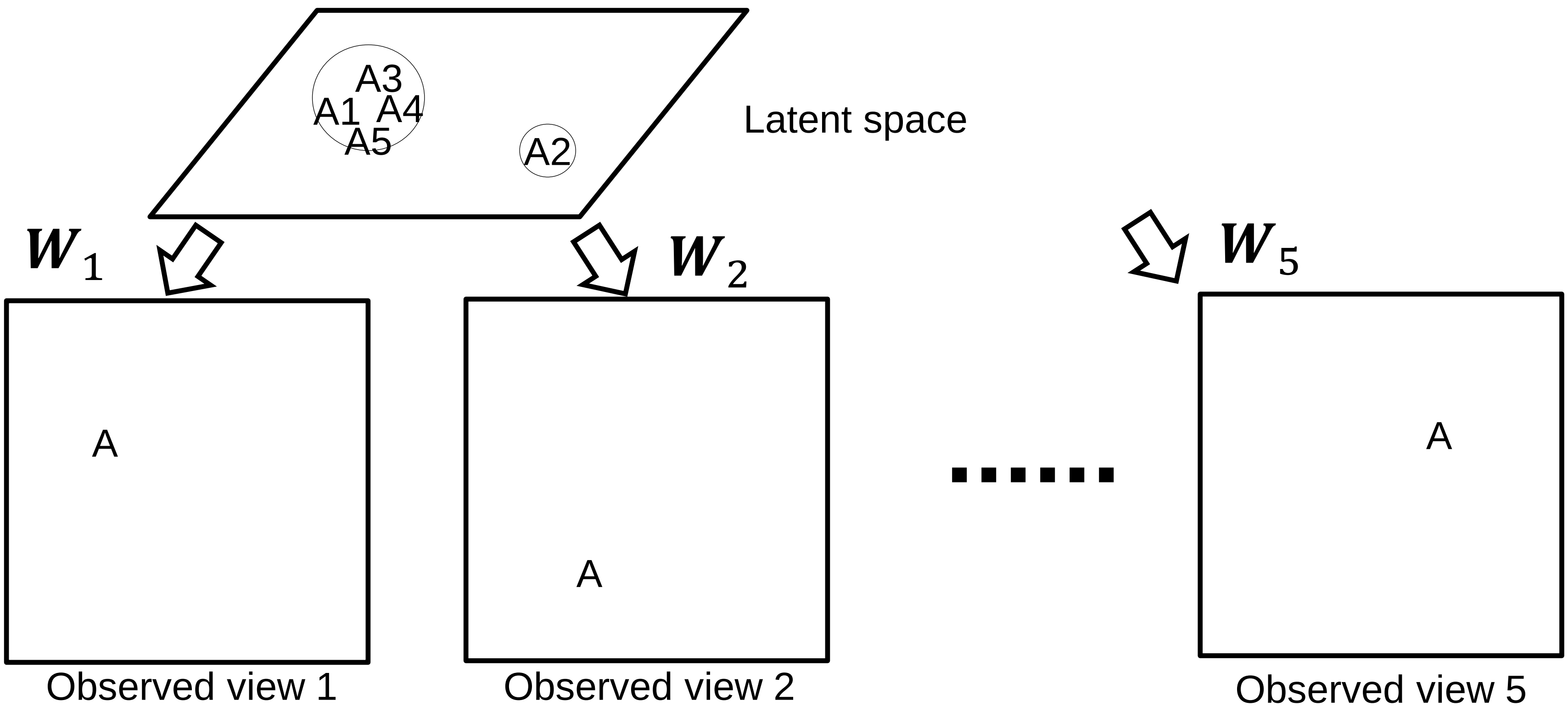
$$v_n = \frac{1}{H} \sum_{h=1}^{H} I({J_n}^{(h)} > 1)$$

#latent vectors of instance n at the h-th iteration after burn-in

indicator function

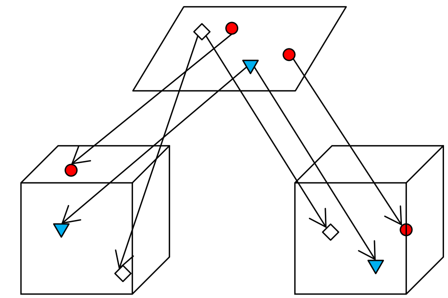
Proposed model

- clustering views for each instance in the latent space

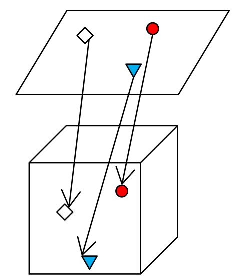


Relation with other latent variable models

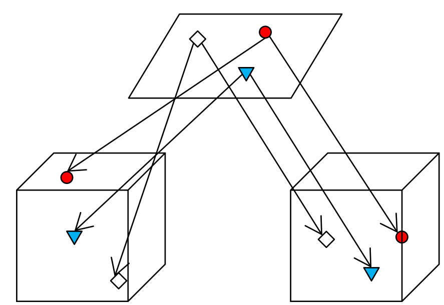
- Proposed model
  - $p(\boldsymbol{x}_{nd}|\boldsymbol{z}_n, \boldsymbol{W}_d, \boldsymbol{\theta}_n, \alpha) = \sum_{j=1}^{\infty} \theta_j N(\boldsymbol{x}_{nd}|\boldsymbol{W}_d \boldsymbol{z}_{nj}, \alpha^{-1}\boldsymbol{I})$



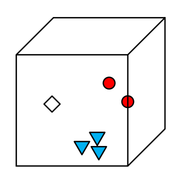
- Probabilistic PCA
  - $p(\boldsymbol{x}_{nd}|\boldsymbol{Z}, \boldsymbol{W}) = N(\boldsymbol{x}_{nd}|\boldsymbol{W}\boldsymbol{z}_{nd}, \alpha^{-1}\boldsymbol{I})$



- Probabilistic CCA
  - $p(\boldsymbol{x}_{nd}|\boldsymbol{Z}, \boldsymbol{W}) = N(\boldsymbol{x}_{nd}|\boldsymbol{W}_d \boldsymbol{z}_{nd}, \boldsymbol{\Sigma})$

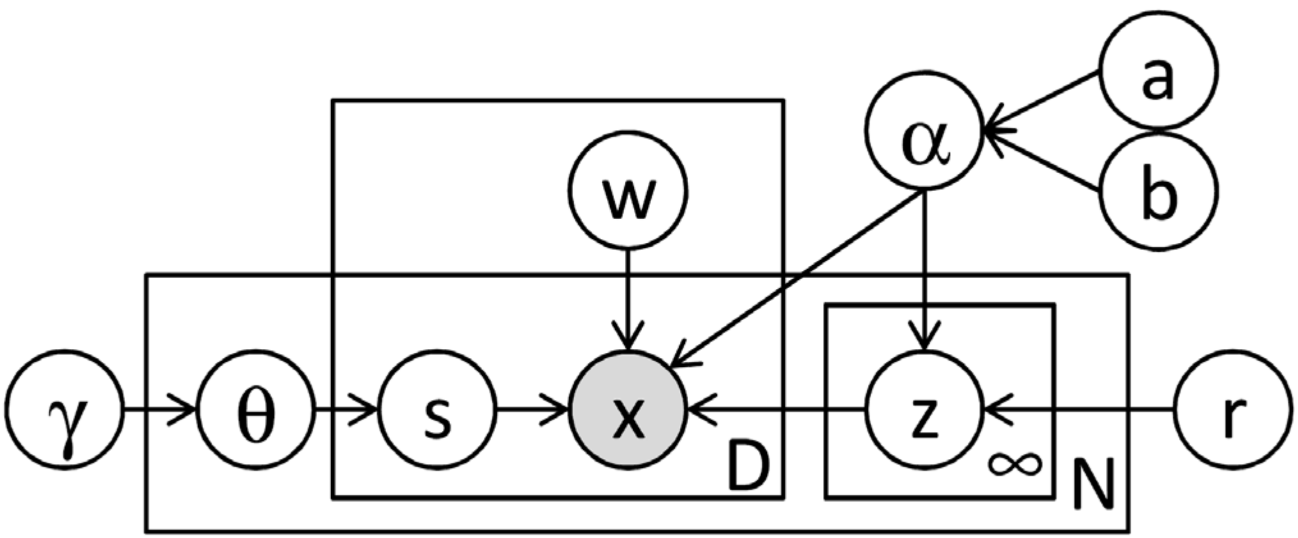


- Infinite Gaussian mixture
  - $p(\boldsymbol{x}_{nd}|\boldsymbol{\mu}, \boldsymbol{\theta}) = \sum_{j=1}^{\infty} \theta_j N(\boldsymbol{x}_{nd}|\boldsymbol{\mu}_j, \alpha^{-1}\boldsymbol{I})$



Generative process

1. Draw a precision parameter $\alpha \sim \text{Gamma}(a, b)$
2. For each instance: $n = 1, \ldots, N$
   (a) Draw mixture weights $\boldsymbol{\theta}_n \sim \text{Stick}(\gamma)$
   (b) For each latent vector: $j = 1, \ldots, \infty$
      i. Draw a latent vector $\boldsymbol{z}_{nj} \sim \mathcal{N}(\mathbf{0}, (\alpha r)^{-1}\boldsymbol{I})$
   (c) For each view: $d = 1, \ldots, D$
      i. Draw a latent vector assignment $s_{nd} \sim \text{Discrete}(\boldsymbol{\theta}_n)$
      ii. Draw an observation vector $\boldsymbol{x}_{nd} \sim \mathcal{N}(\boldsymbol{W}_d \boldsymbol{z}_{ns_{nd}}, \alpha^{-1}\boldsymbol{I})$



Inference based on stochastic EM

- Analytically integrate out the latent vectors Z, mixture weights Θ, precision α
- E-step: collapsed Gibbs sampling of latent vector assignment s for each view of each instance $\ell = (n, d)$

$$p(s_\ell = j | \boldsymbol{X}, \boldsymbol{S}_{\backslash \ell}, \boldsymbol{W}, a, b, r, \gamma) \propto \frac{p(s_\ell = j, \boldsymbol{S}_{\backslash \ell} | \gamma)}{p(\boldsymbol{S}_{\backslash \ell} | \gamma)} \cdot \frac{p(\boldsymbol{X} | s_\ell = j, \boldsymbol{S}_{\backslash \ell}, \boldsymbol{W}, a, b, r)}{p(\boldsymbol{X}_{\backslash \ell} | \boldsymbol{S}_{\backslash \ell}, \boldsymbol{W}, a, b, r)}$$

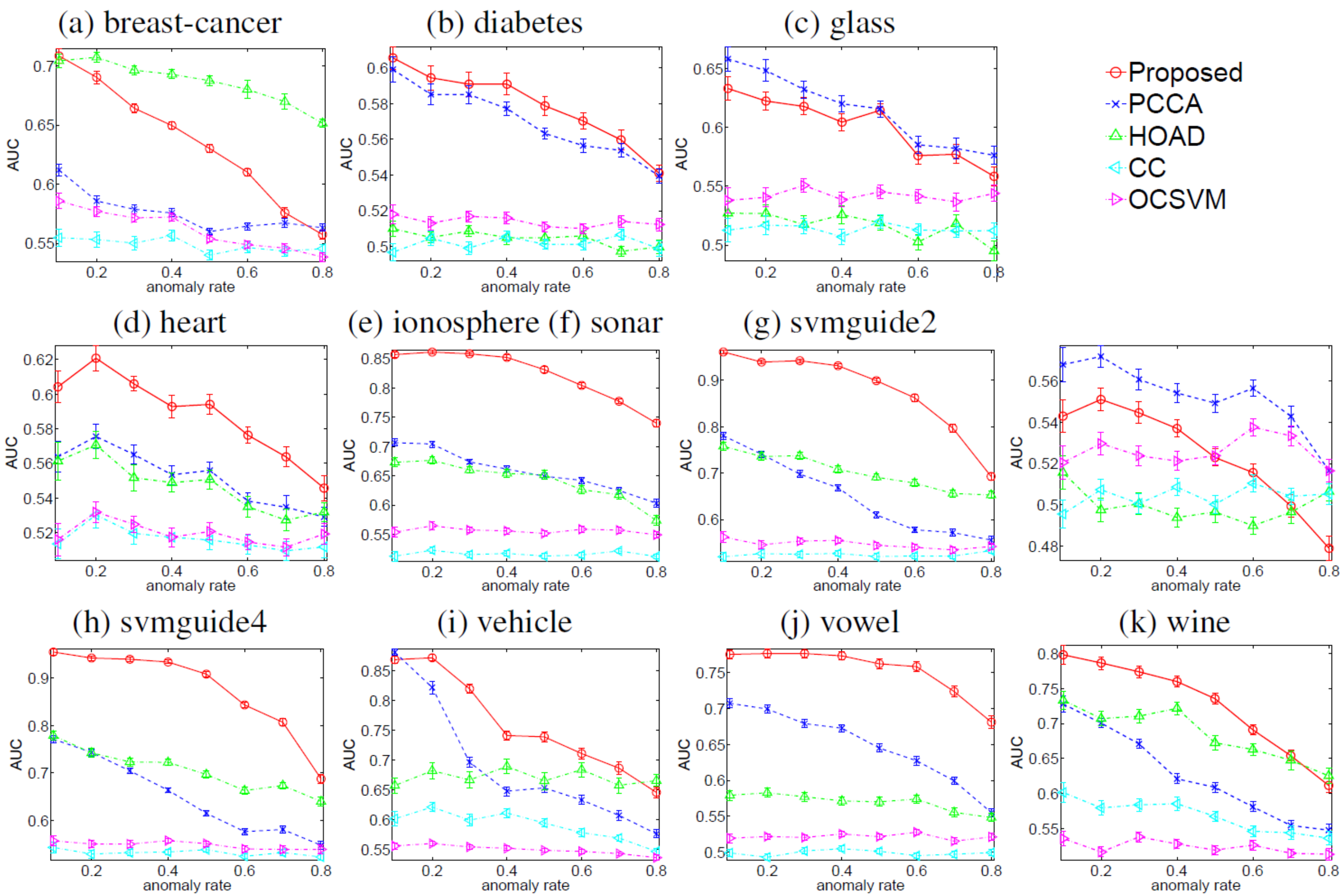
- M-step: maximum joint likelihood estimation of projection matrices W

$$\boldsymbol{W}_d = \Big(\frac{a'}{b'} \sum_{n=1}^{N} \boldsymbol{x}_{nd} \boldsymbol{\mu}_{n s_{nd}}^\top \Big) \Big(\sum_{n=1}^{N} \sum_{j=1}^{J_n} \boldsymbol{C}_{nj} + \frac{a'}{b'} \sum_{n=1}^{N} \boldsymbol{\mu}_{n s_{nd}} \boldsymbol{\mu}_{n s_{nd}}^\top \Big)^{-1}$$

Experiments

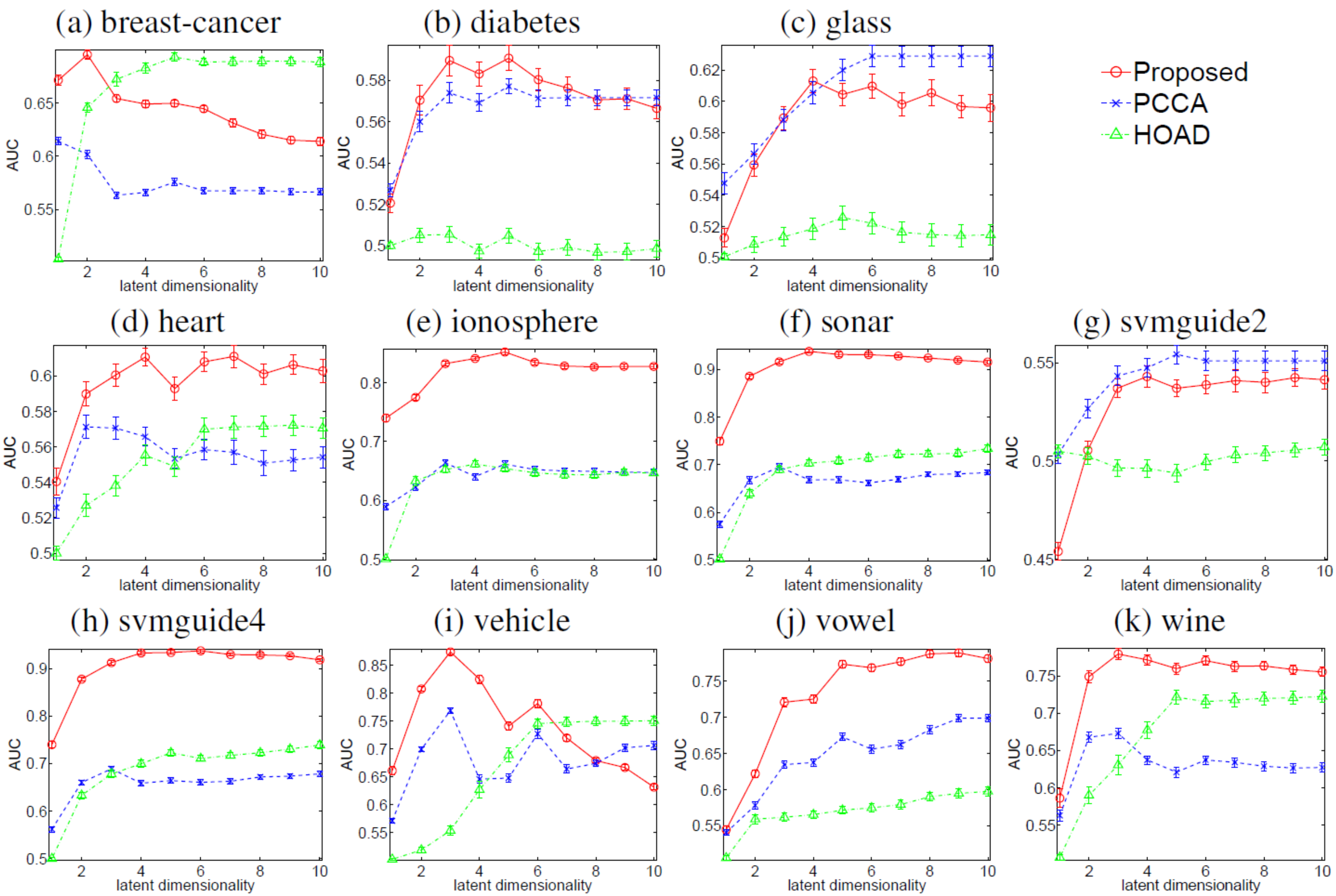
- Data
  - 11 data sets from LIBSVM data
  - generated multiple views by randomly splitting the features
  - anomalies were added by swapping views of two randomly selected instances
- Comparing methods
  - PCCA: probabilistic canonical correlation analysis
  - HOAD: horizontal anomaly detection
  - CC: consensus clustering based anomaly detection
  - OCSVM: one-class support vector machine

Multi-view anomaly detection with different anomaly rate



The proposed model achieved the best with 8 of the 11 data sets

Multi-view anomaly detection with different latent dimensionality



MovieLens data analysis

instance: movie, view1: user list who rated, view2: genre

Title	Score	Title	Score
The Full Monty	0.98	Star Trek VI	0.04
Liar Liar	0.93	Star Trek III	0.04
The Professional	0.91	The Saint	0.04
Mr. Holland's Opus	0.88	Heat	0.03
Contact	0.87	Conspiracy Theory	0.03

high anomaly score movies   low anomaly score movies

'The Full Monty' and 'Liar Liar' were 'Comedy' genre. They are rated by not only users who likes 'Comedy', but also who likes 'Romance' and 'Action-Thriller'.
'The Professional' was anomaly because it was rated by two different user groups, where a group prefers 'Romance' and the other prefers 'Action'.
Since 'Star Trek' series are typical Sci-Fi and liked by specific users, its anomaly score was low.

Conclusion

- We proposed a generative model approach for multi-view anomaly detection, which finds instances that have inconsistent views.
- In the experiments, we confirmed that the proposed model could perform much better than existing methods for detecting multi-view anomalies
- Future work
  - nonlinear projection using Gaussian processes or deep neural nets