2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

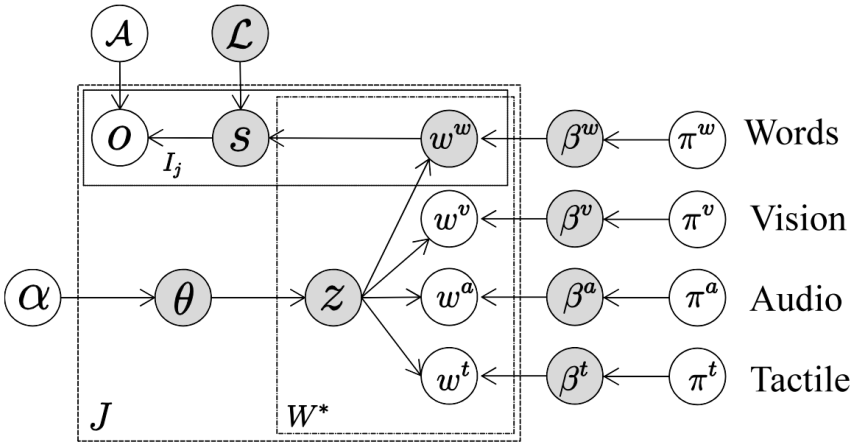
Serket: A Framework for Construction of Multimodal Learning Models

The University of Electro-Communications
Tomoaki Nakamura

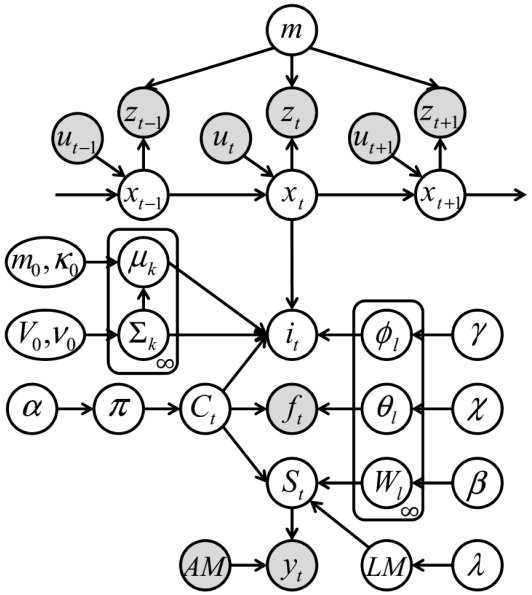
978-1-7281-6211-9/20/$31.00 ©2020 IEEE

Background

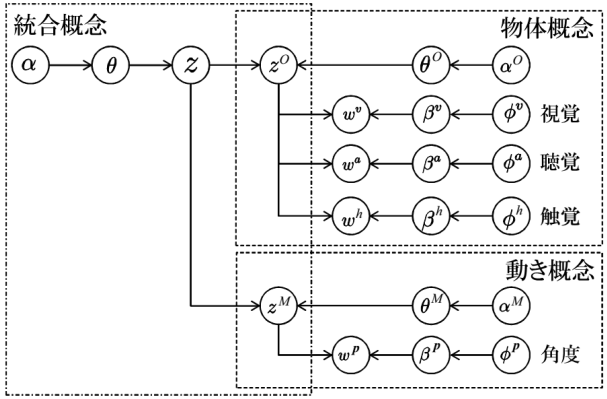
- Our group has developed many multimodal learning models based on probabilistic models



Learning object and language [Nakamura+ 15]



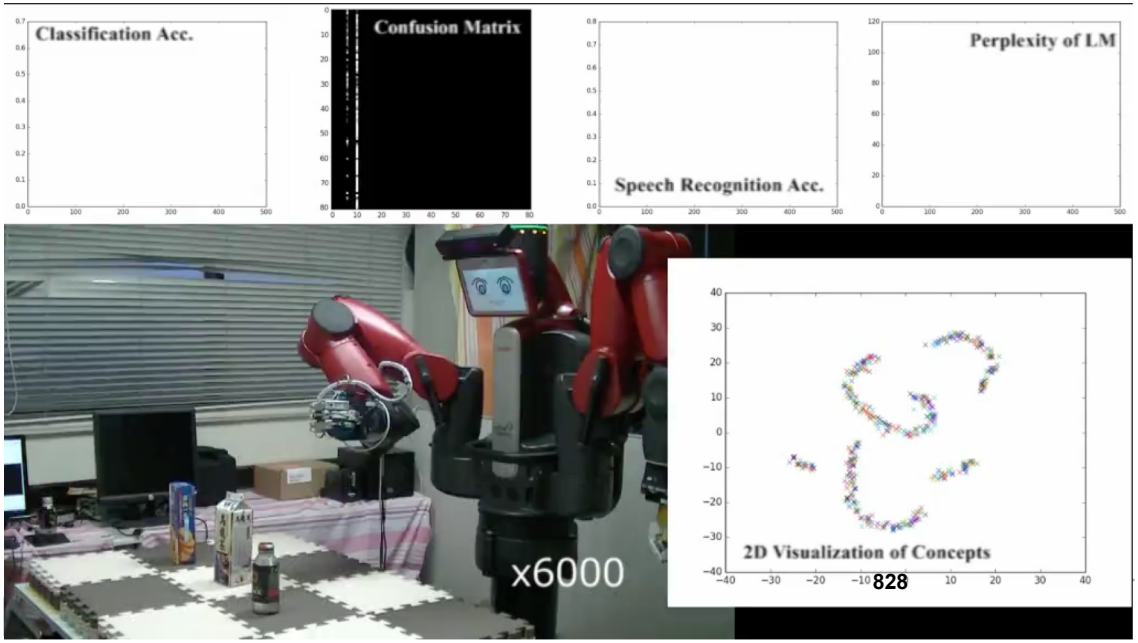
Place concept formation [Taniguchi+ 17]

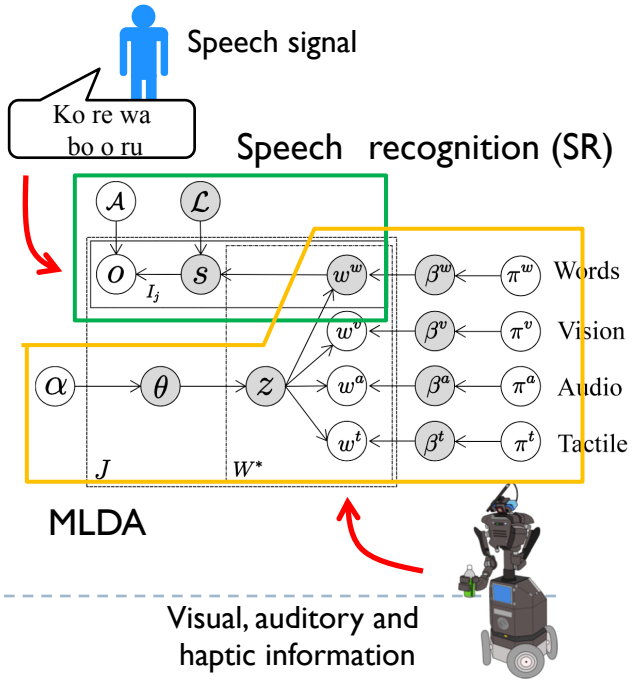


Formation of integrated concepts with object and motion [Attamimi+ 14]

Learning object concepts and language model

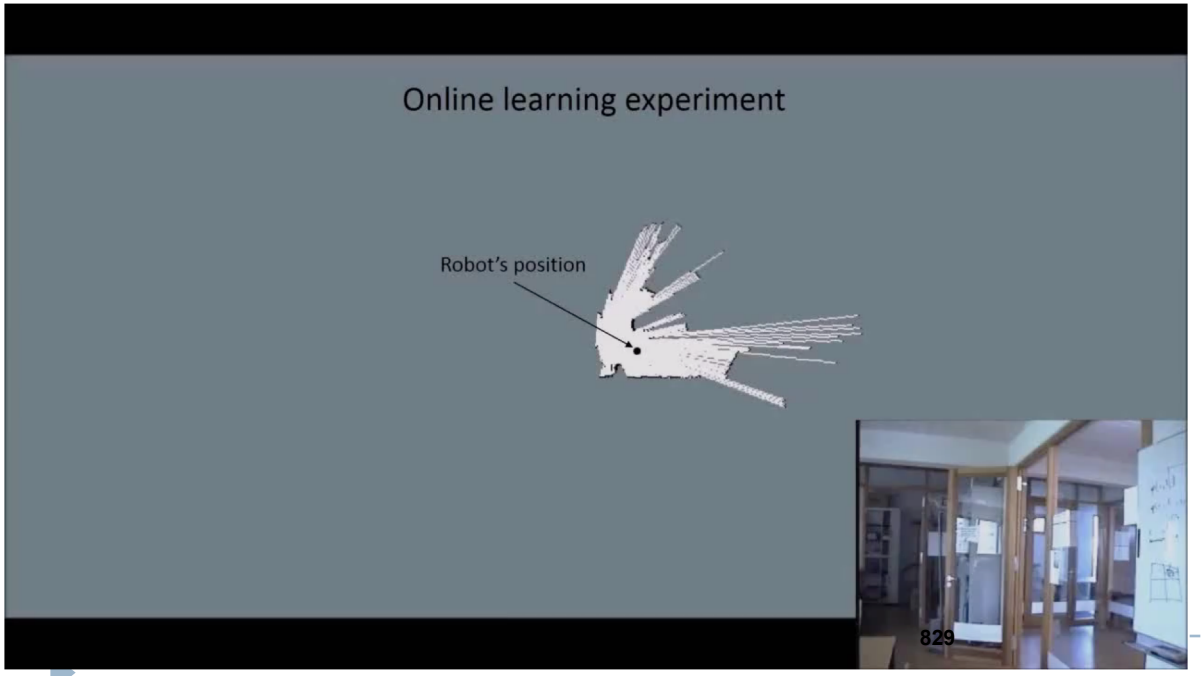
- Multimodal information is classified into categories (MLDA)
  - Robot obtain multimodal (visual, auditory and haptic) information by observing, grasping and shaking objects
  - User teaches object features by speech
- We assumed robot does not have predefined language knowledge
  - ➡ Parameters of speech recognition (SR) are learned simultaneously

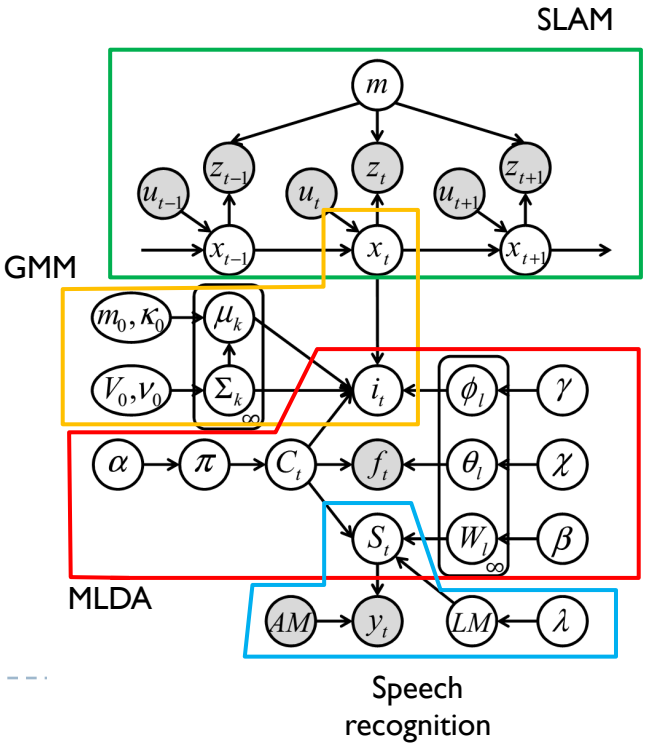




Spatial Concept Acquisition [Taniguchi+ 2017]

- Taniguchi et al. proposed more complicated model
- Robot builds a map using SLAM and simultaneously learns:
  - space name and appearance (MLDA), space region (GMM), parameters of speech recognition (LM)

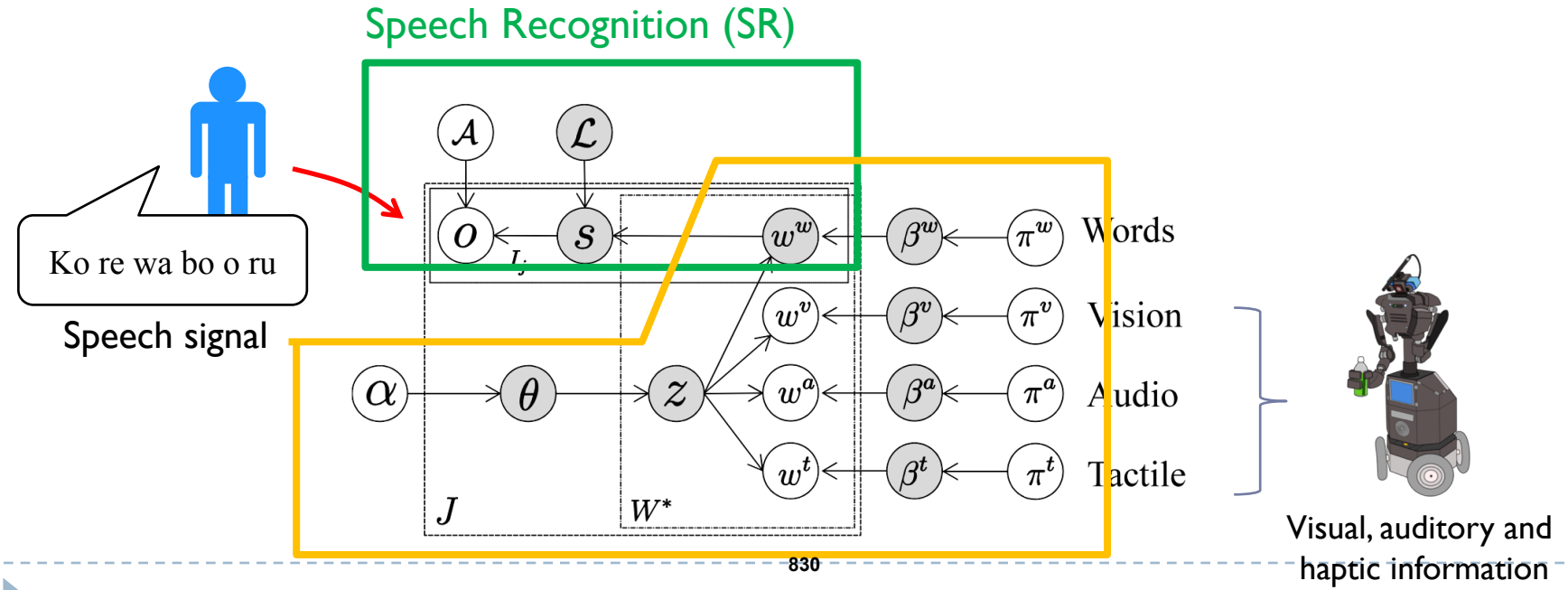




Multimodal Learning=Complementary Learning

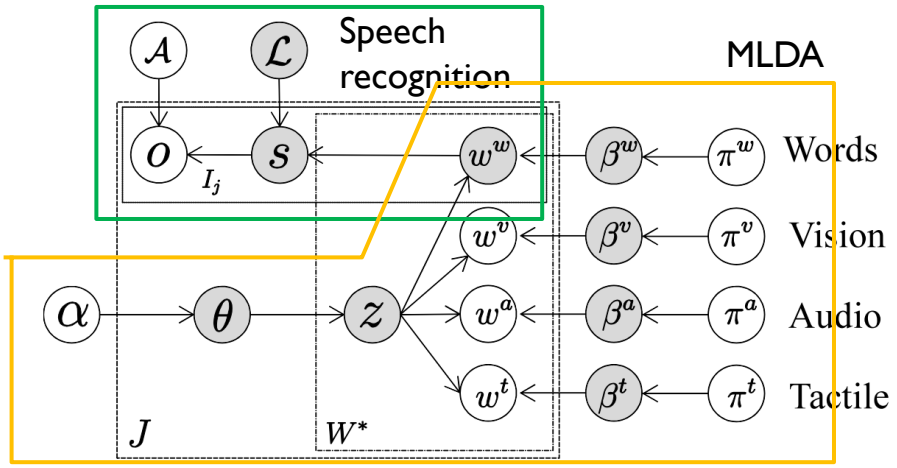
- This model is constructed by connecting two models:
  - Speech recognition and clustering (latent Dirichlet allocation)
- Shared variable w^w is determined with mutual influence
  - Possibility that speech $\boldsymbol{o}$ is recognized as w^w
  - Possibility that w^w is co-occurred with category z

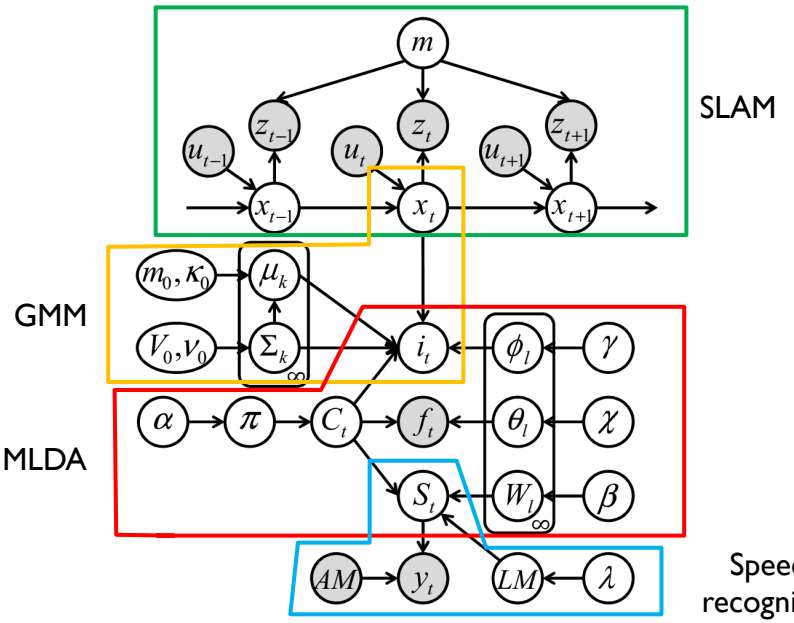
complementary learning

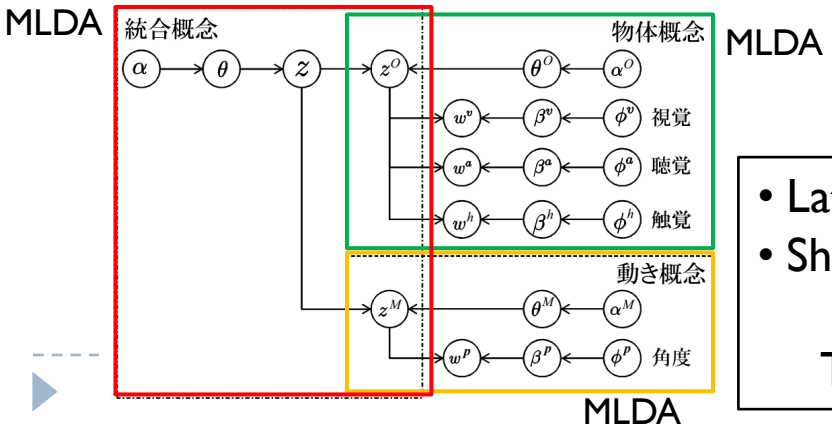


Background

- These models are also constructed by connecting small-scale models







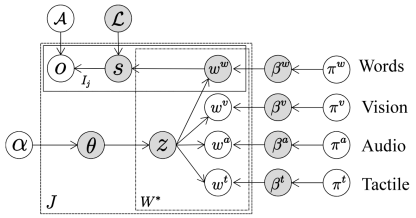
- Latent variables are shared with two models
- Shared variable is determined with mutual influence

The parameters are optimized complementarily

Problem in Constructing Models

- These models have complicated structure
- To realize human-like learning models, much more complicated models are required

Formulate the learning system



Drive equations for parameter inference

$$P(x_{new}|\boldsymbol{X}_k) = \frac{P(x_{new},\boldsymbol{X}_k)}{P(\boldsymbol{X}_k)} = \frac{Z(\{\alpha'_{k0},\cdots,\alpha'_{k,w=x_{new}}+1,\cdots,\alpha'_{kW}\})}{Z(\boldsymbol{\alpha}'_k)}$$
$$= \frac{\Gamma(\alpha'_{k0})\cdots\Gamma(\alpha'_{k,w=x_{new}}+1)\cdots\Gamma(\alpha'_{kW})}{\Gamma(\sum_w^W \alpha'_{0w}+1)} \frac{\Gamma(\sum_w^W \alpha'_{0w})}{\prod_w^W \Gamma(\alpha'_{0w})}$$
$$= \frac{\alpha'_{k,w=x_{new}} \prod_w^W \Gamma(\alpha'_{kw}))}{\sum_w^W \alpha'_{0w}\Gamma(\sum_w^W \alpha'_{0w})} \frac{\Gamma(\sum_w^W \alpha'_{0w})}{\prod_w^W \Gamma(\alpha'_{0w})}$$
$$= \frac{\alpha'_{k,w=x_{new}}}{\sum_w^W \alpha'_{0w}} = \frac{N_{k,w=x_{new}}+\alpha'_{k,w=x_{new}}}{\sum_w^W N_{kw}+\alpha_{0w}}$$

Implement it with the programing

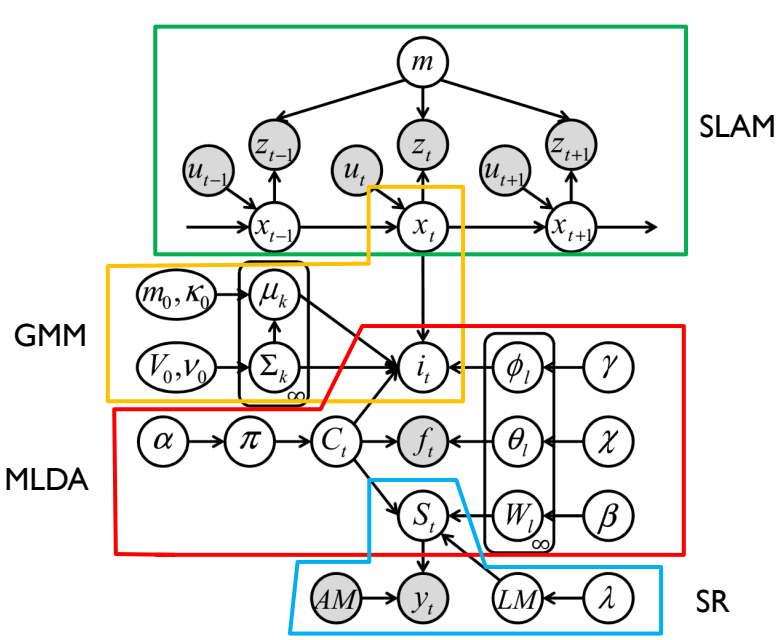
```
def forward_filtering(self, d ):
    T = len(d)
    a = np.zeros( (len(d), self.MAX_LEN, self.numclass) ) - 1.0e-
    valid = np.zeros( (len(d), self.MAX_LEN, self.numclass) ) # i
    z = np.ones( T ) # 正規化定数

    for t in range(T):
        for k in range(self.MIN_LEN,self.MAX_LEN,self.SKIP_LEN):
            if t-k<=self.MIN_LEN:
                break

            segm = d[t-k:t+1]
            for c in range(self.numclass):
                out_prob = self.calc_emission_logprob( c, segm )
                foward_prob = 0.0

                # 遷移確率
```

This part becomes more difficult

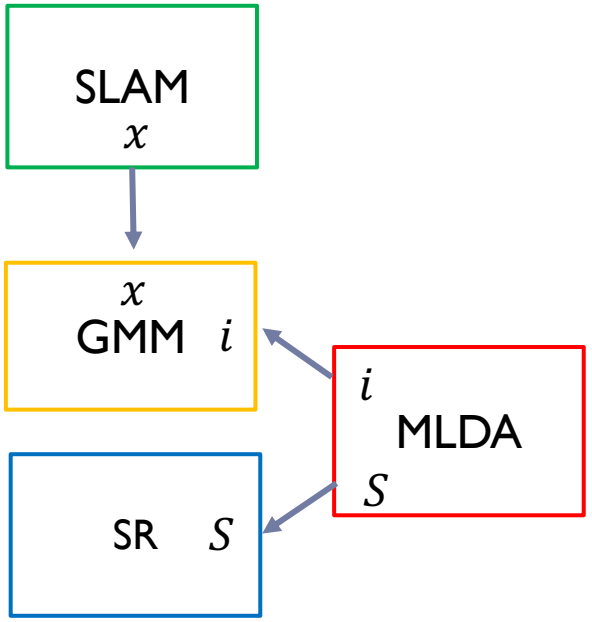
- Framework to easily construct models is required

SERKET

- To easily construct multimodal learning models, we have proposed the framework **SERKET**



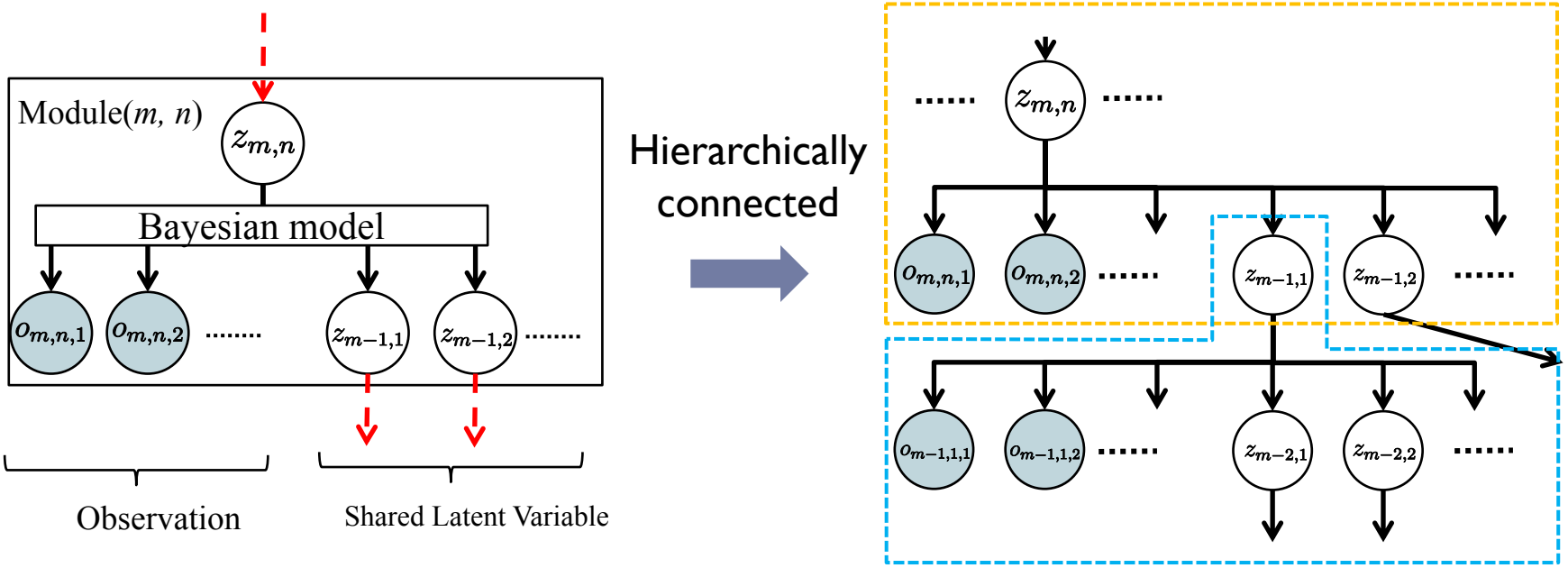
SERKET
- Modularizing
- Connecting
- Inference



- Shared latent variables are optimized with mutual influence on each model

Modules in SERKET

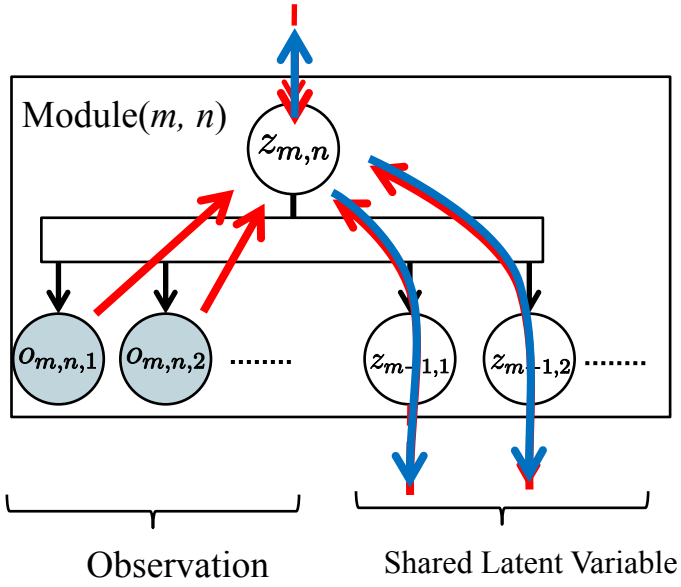
- Each module has observations and latent variables
- Models are constructed by connecting shared latent variables of modules hierarchically



- Parameters are estimated by communication between modules while programmatic independence maintains

Estimation of shared latent variables

- Parameters are estimated by exchanging messages

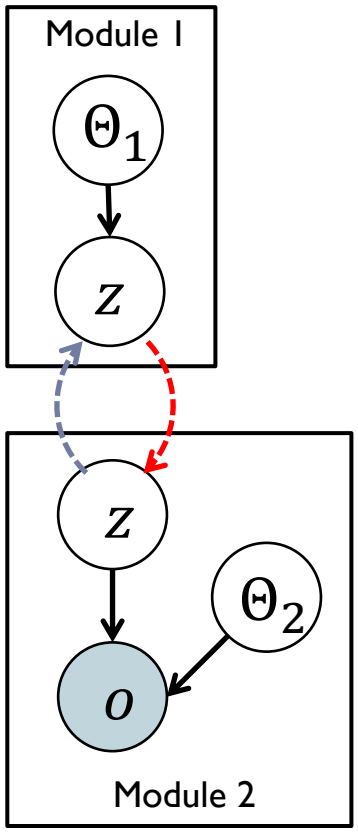


1. Receive messages from other modules and latent variable $z_{m,n}$ and parameters are updated
2. New messages are sent to other modules based on updated parameters

- By executing this procedure sequentially in each module, the parameters are optimized mutually
- Currently, two methods are implemented for message exchange

Parameter optimization 1

- Message Passing (MP) Approach

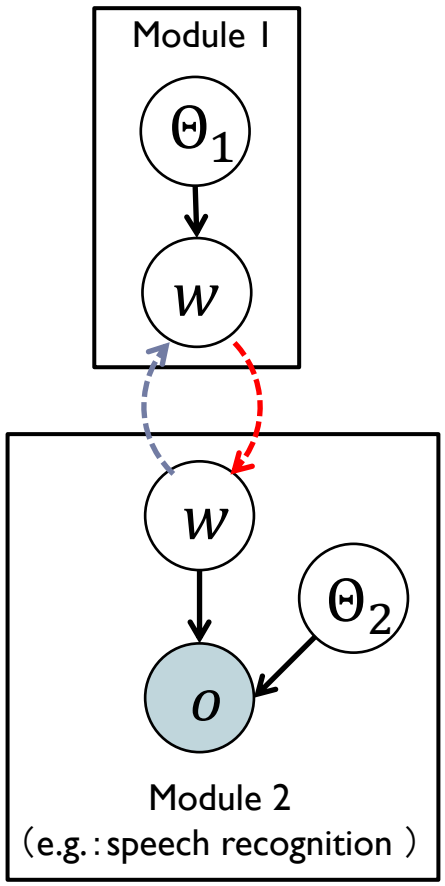


- Assume that z is determined with mutual influence : $z \sim P(z|\Theta_1, \Theta_2, o)$
  $\propto P(z|\Theta_1)P(z|\Theta_2, o)$
- Module 1→Module 2:
  - Module 1 sends $P(z|\Theta_1)$ to Module 2
  - Module 2 updates Θ_2 using $P(z|\Theta_1)$
- Module 2→Module 1:
  - Module 2 sends $P(z|o, \Theta_2)$ to Module 1
  - Module 1 updates Θ_1 using $P(z|o, \Theta_2)$

Θ_1 : Parameters of Module 1
Θ_2 : Parameters of Module 2
z : Shared latent variables
o : Observation

Parameter optimization 2

- Sampling Importance Resampling (SIR) approach



- In the case that w has a large number of possibilities such as speech recognition results ➡ Monte Carlo approximation
- Module 2 generates samples w_n and sends them to Module 1

$$w_n \sim P(w|\Theta_2, o)$$

- Module 1 resamples based on $P(w|\Theta_1)$ and sends selected samples w^* to Module 2
- Parameters are updated using selected w^*

Implementation Examples

- We confirm that integrated models by SERKET improve their performance
- Used modules:
  - Variational Auto-Encoder (VAE)
  - Gaussian Mixture Model (GMM)
  - Multimodal Latent Dirichlet Allocation (MLDA)
  - Markov Model (MM)
  - Speech Recognition (SR)
- Multimodal Dataset :
  - Image : MNIST
  - Speech : Spoken Arabic Digit Dataset

Implementation Examples

- Five examples
  1. VAE+GMM
  2. VAE+GMM+MLDA
  3. VAE+GMM+MLDA+MM
  4. Model for language acquisition by robots
  5. Model for learning object feature extractor by robots

We share Jupyter notebook that you can execute on the Google Collaboratory

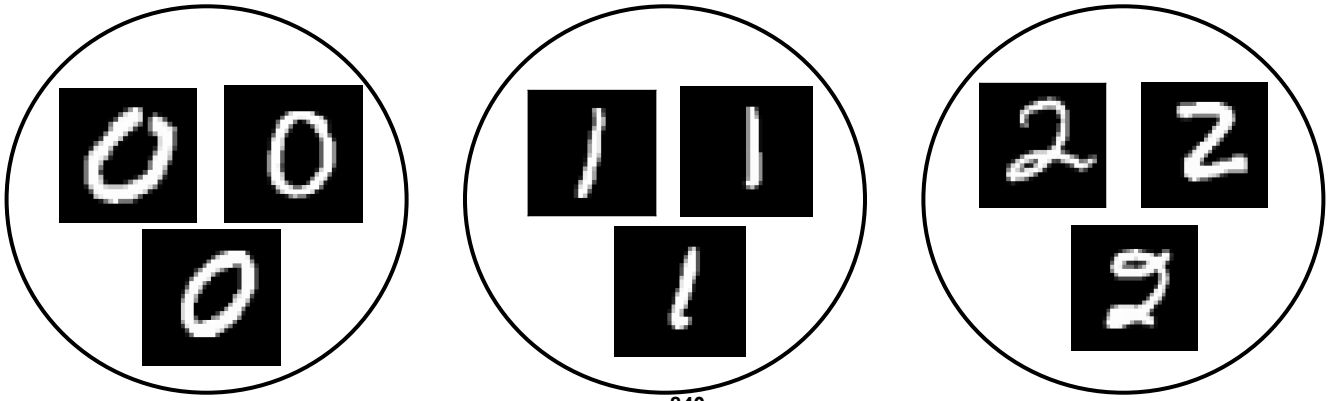
http://iros20.naka-lab.org/

Example1：VAE+GMM

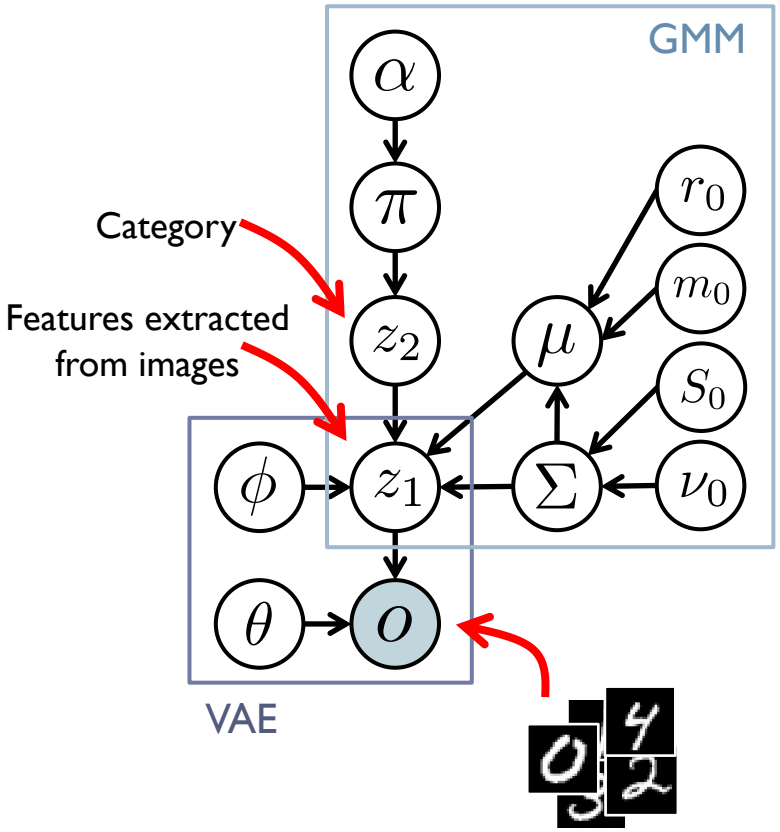
- Unsupervised image classification

Image data: MNIST (784 dims)

- Dimensional compression by VAE (18 dims)
- Classification by GMM



Example1：VAE+GMM




```
1  import serket as srk
2  import vae
3  import gmm
4  import numpy as np
5
6   # Load observations
7  obs1 = srk.Observation(np.loadtxt("data.txt"))
8  category = np.loadtxt("category.txt")
9
10  # Define modules
11 vae1 = vae.VAE(18, itr=200, batch_size=500)
12 gmm1 = gmm.GMM(10, category=category)
13
14  # Define connection between modules
15 vae1.connect(obs1)
16 gmm1.connect(vae1)
17
18  # Optimize the parameters
19 for i in range(5):
20     vae1.update()
21     gmm1.update()
```

Source code

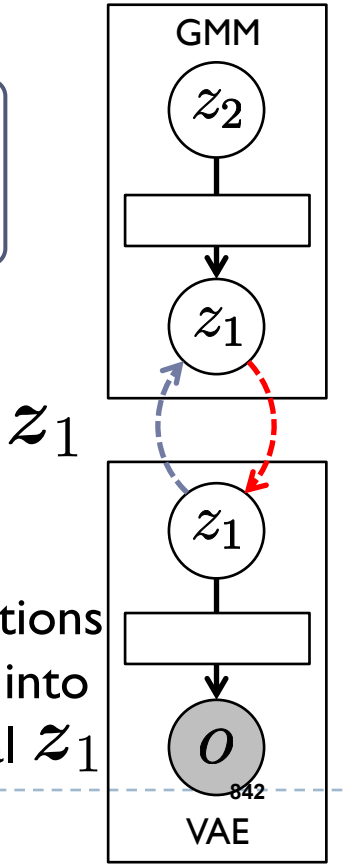
Example1: Mutual learning of VAE and GMM

- Modified evidence lower bound (ELBO) of VAE

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{o}) = -D_{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}_1|\boldsymbol{o}) \| \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_1|\boldsymbol{o})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{o}|\boldsymbol{z}_1)]$$

Standard VAE: $\boldsymbol{\mu} = \boldsymbol{0}$

Latent space suitable
for classification can be learned
with **mutual influence**



Compute $\boldsymbol{\mu}$ of cluster

$\boldsymbol{z}_1$ $\boldsymbol{\mu}$

784 dimensional observations
are compressed into
18 dimensional $\boldsymbol{z}_1$

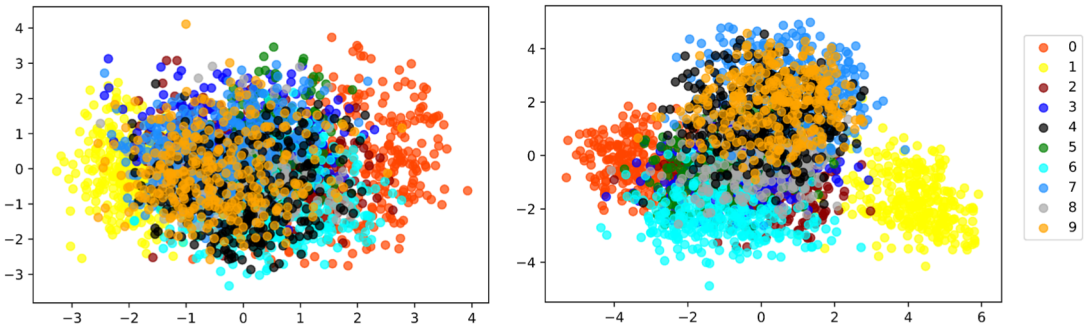
$\boldsymbol{z}_1$ is estimated using modified ELBO
and received $\boldsymbol{\mu}$

Example1: Classification Results

Adjusted Rand Index (ARI)

Mutual learning	average	best
-	0.477	0.478
✓	0.503	0.568

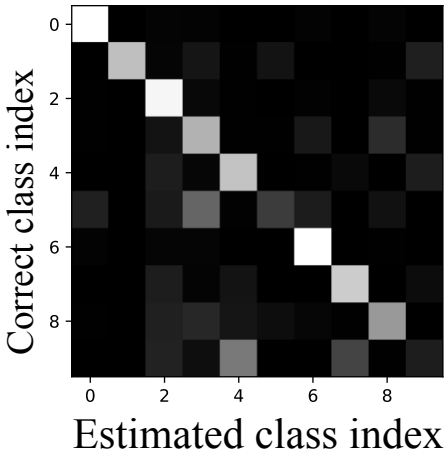
※ average: Average of 10 trials
best: The best result in 10 trials



w/o mutual learning　　w/ mutual learning

Latent variables compressed by PCA
(18 dims→2 dims)

Confusion Matrix



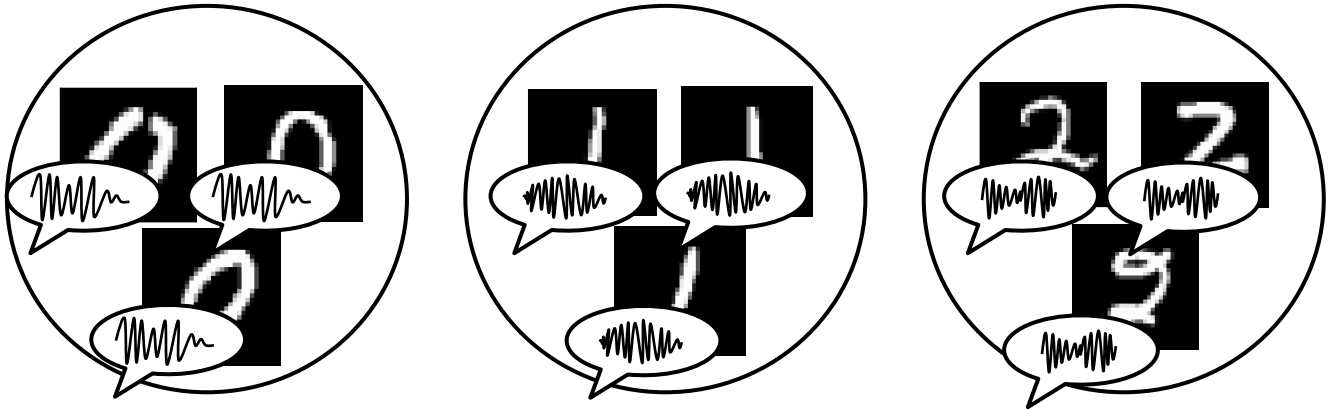
- **Latent space suitable for classification** was learned with mutual influence
- Classification accuracy increased

Example2: VAE+GMM+MLDA

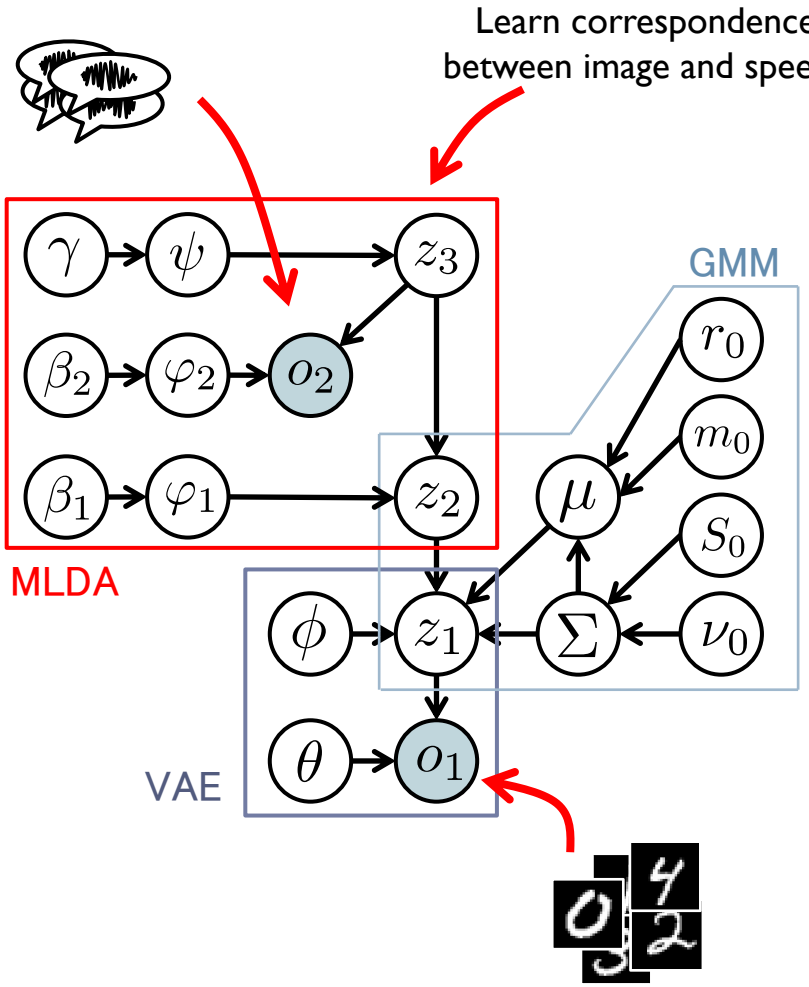
- Unsupervised classification of image and speech

Pairwise dataset of image and speech

- Multimodal data is classified in unsupervised manner



Example 2: VAE+GMM+MLDA




```
1  import serket as srk
2  import vae
3  import gmm
4  import mlda
5  import numpy as np
6
7   # Load observations
8  obs1 = srk.Observation(np.loadtxt("data1.txt")) # 画像
9  obs2 = srk.Observation(np.loadtxt("data2.txt")) # 音声
10 category = np.loadtxt("category.txt")
11
12  # Define modules
13 vae1 = vae.VAE(18, itr=200, batch_size=500)
14 gmm1 = gmm.GMM(10, category=category)
15 mlda1 = mlda.MLDA(10, category=category)
16
17  # Define connection between modules
18 vae1.connect(obs1)
19 gmm1.connect(vae1)
20 mlda1.connect(obs2, gmm1)
21
22  # Optimize the parameters
23 for i in range(5):
24     vae1.update()
25     gmm1.update()
26     mlda1.update()
```

Source code

Example 2: Classification Results

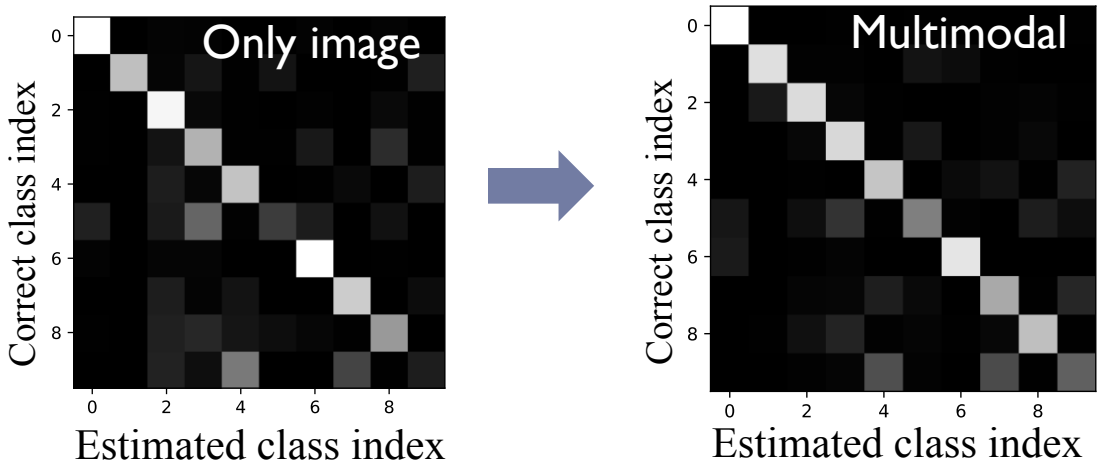
Adjusted Rand Index (ARI)

Mutual learning	average	best
-	0.604	0.638
✓	0.637	0.735

Classification accuracy improved by **multimodal learning** using images and speech

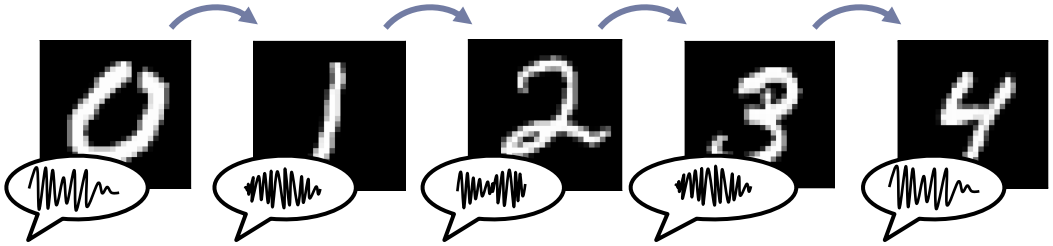
※ average: Average of 10 trials
best: The best result in 10 trials

Confusion Matrix



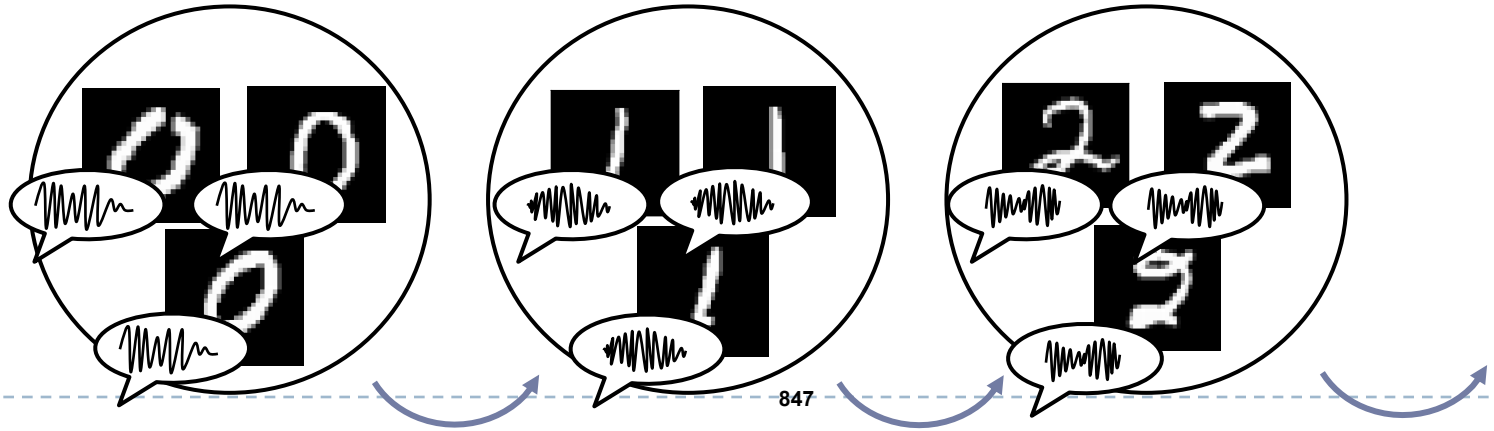
Example 3: VAE+GMM+MLDA+MM

- Category and their sequence rules are learned

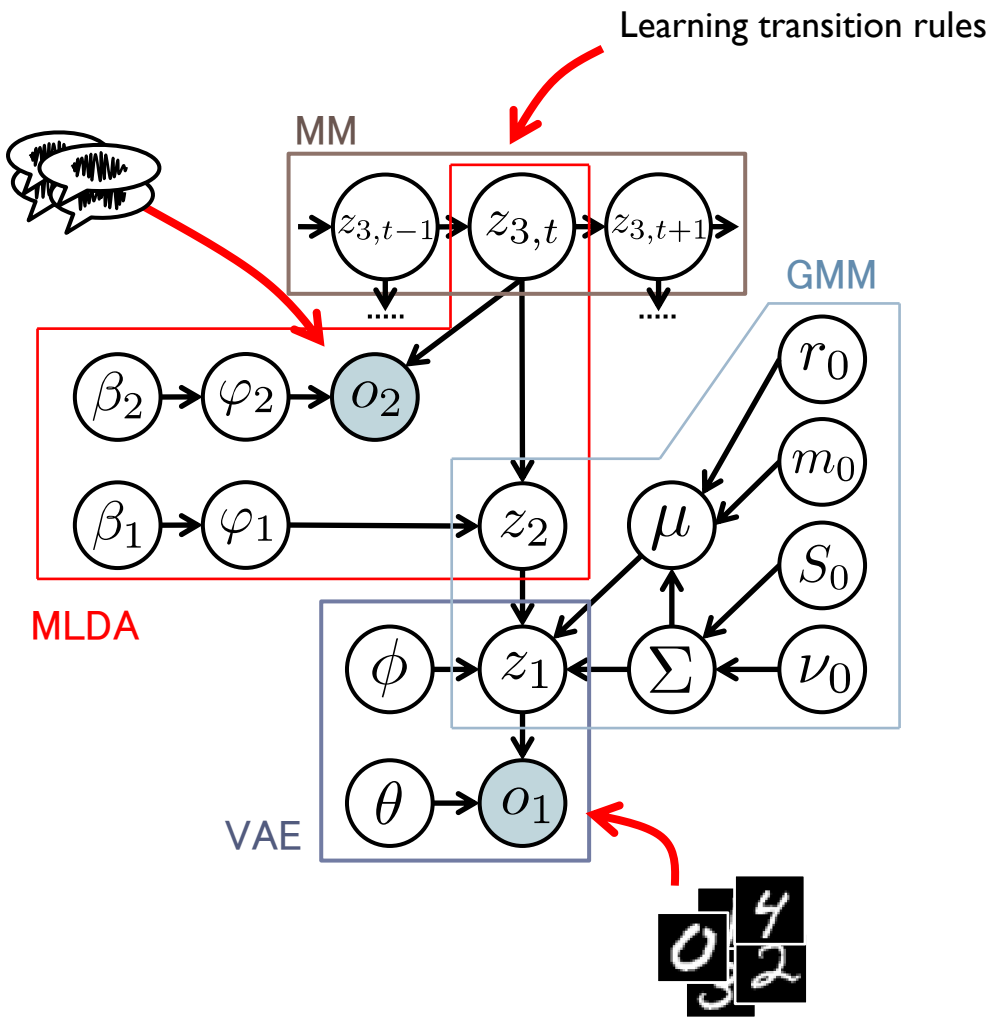


Arranged pairwise data in ascending order

- Unsupervised classification of multimodal data
- Learning their transition rules



Example 3: VAE+GMM+MLDA+MM




```
import serket as srk
import vae
import gmm
import mlda
import mm
import numpy as np

# Load observations

obs1 = srk.Observation(np.loadtxt("data1.txt")) # 画像
obs2 = srk.Observation(np.loadtxt("data2.txt")) # 音声
category = np.loadtxt("category.txt")

# Define modules

vae1 = vae.VAE(18, itr=200, batch_size=500)
gmm1 = gmm.GMM(10, category=category)
mlda1 = mlda.MLDA(10, category=category)
mm1 = mm.MarkovModel()

# Define connection between modules

vae1.connect(obs1)
gmm1.connect(vae1)
mlda1.connect(obs2, gmm1)
mm1.connect(mlda1)

# Optimize the parameters

for i in range(5):
    vae1.update()
    gmm1.update()
    mlda1.update()
    mm1.update()
```

Suorce code

Example 3: Classification Results

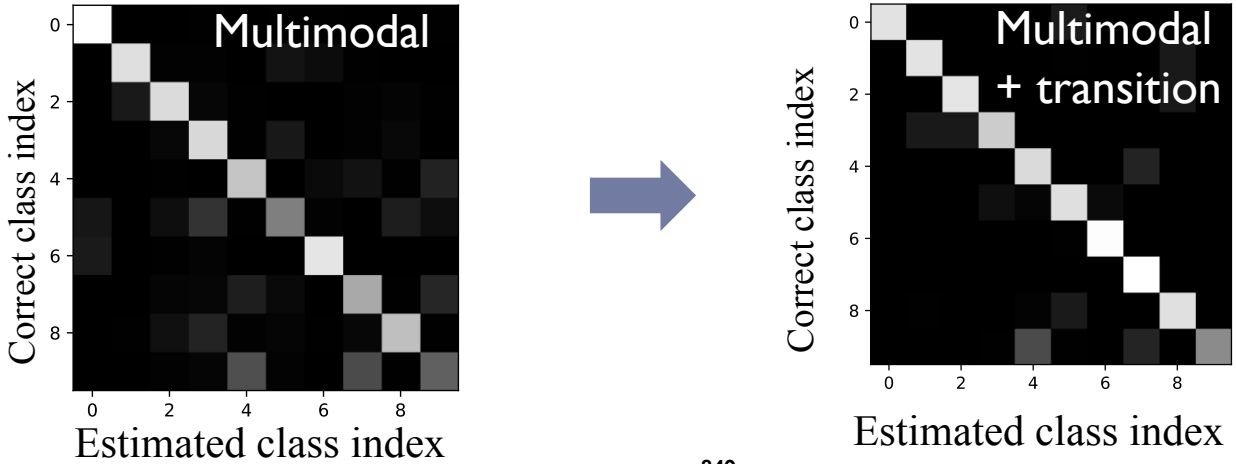
Adjusted Rand Index (ARI)

Mutual learning	average	best
-	0.575	0.524
✓	0.834	0.980

Classification accuracy significantly improved by learning transition rules

※ average: Average of 10 trials
best: The best result in 10 trials

Confusion Matrix

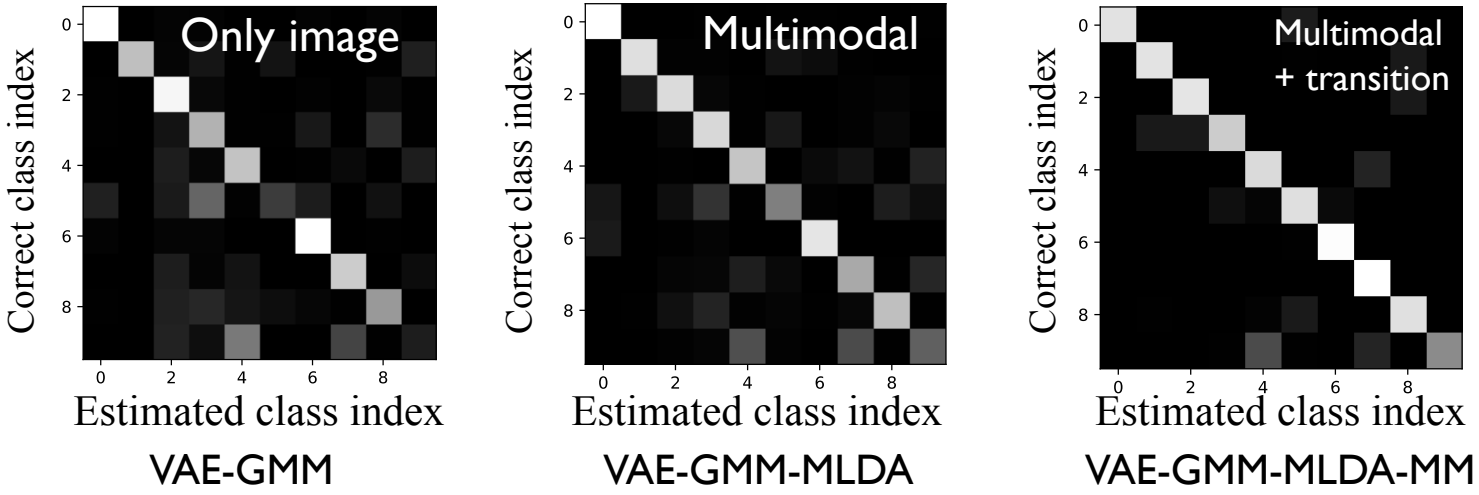


Conclusion of Example 1,2 and 3

Adjusted Rand Index (ARI)

	VAE+GMM	VAE+GMM+MLDA	VAE+GMM+MLDA+MM
average	0.503	0.637	0.834

Confusion Matrix

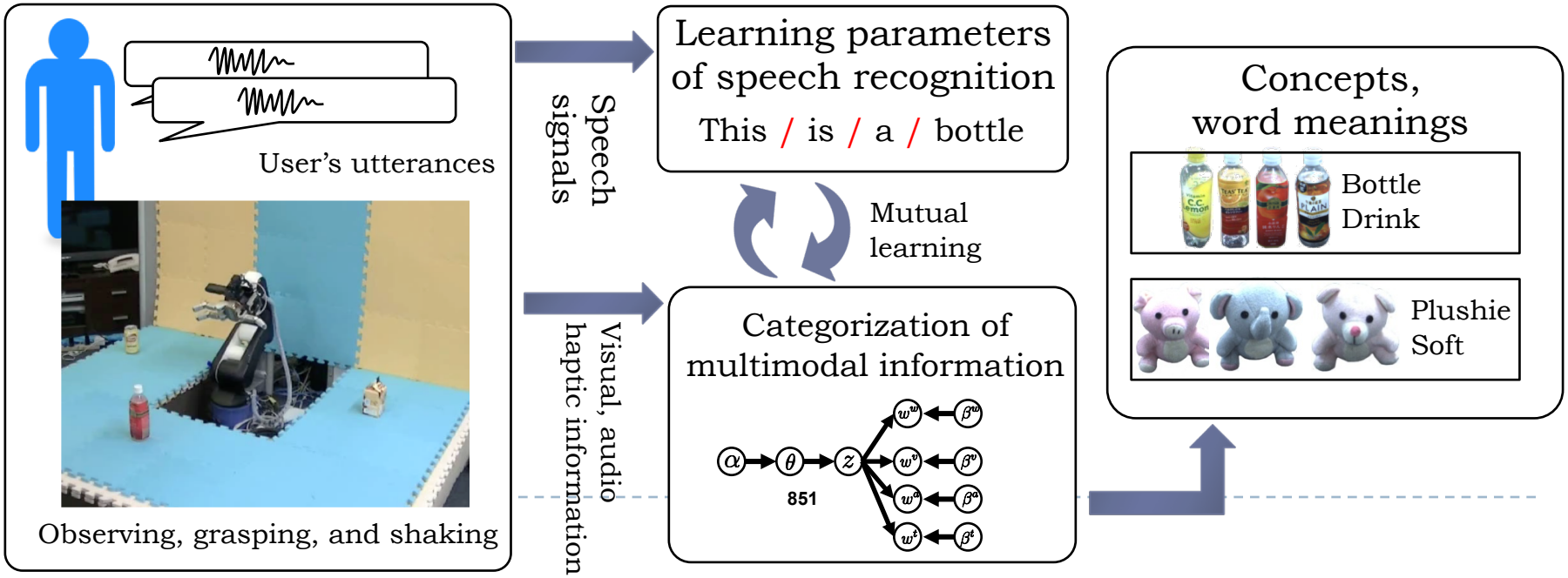


Complementary learning is realized by using SERKET

Example 4: Model for language acquisition

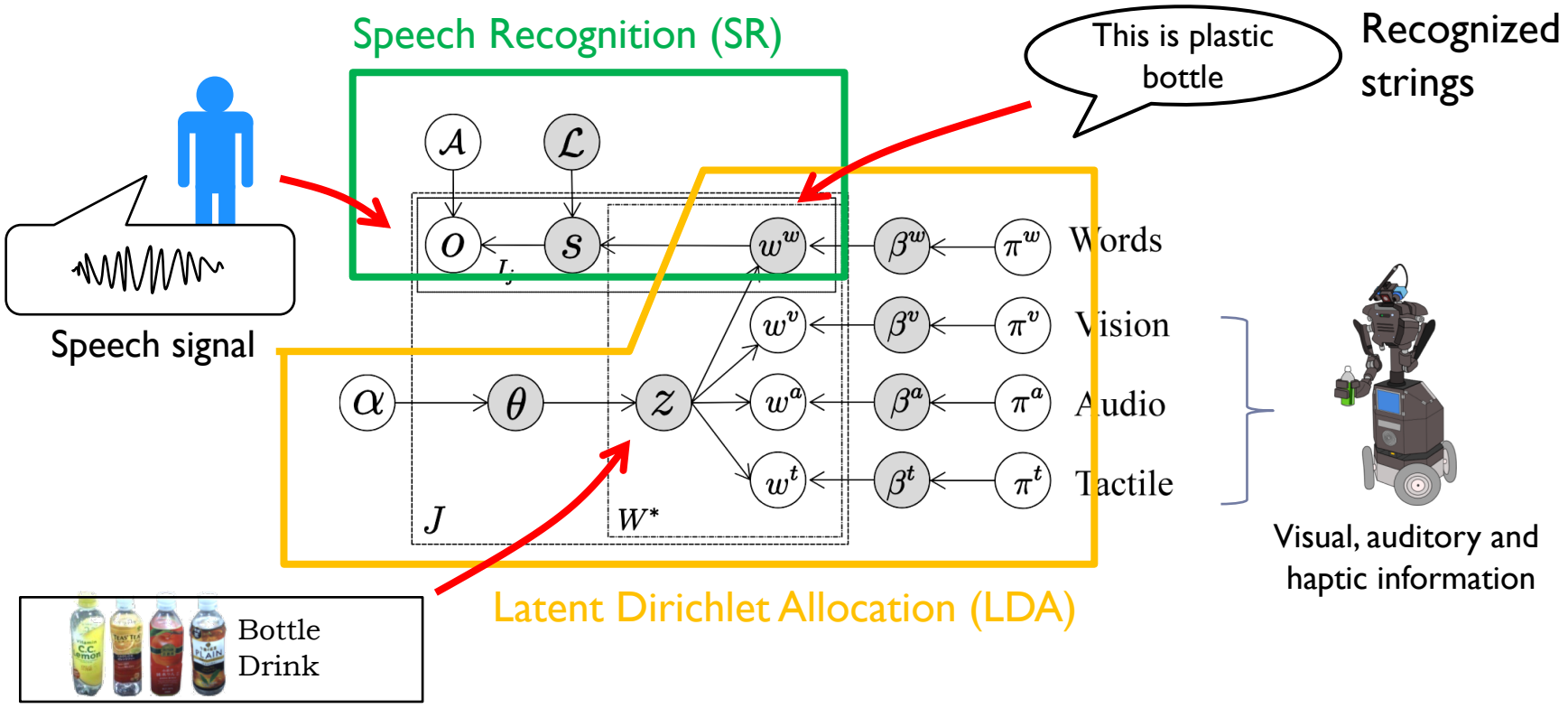
- Unsupervised classification of multimodal dataset
  - Robot obtain multimodal (visual, auditory and haptic) information by observing, grasping and shaking objects
  - User teaches object features by speech
- We assumed robot does not have language knowledge

  ➡ Parameters of speech recognition (SR) are learned simultaneously



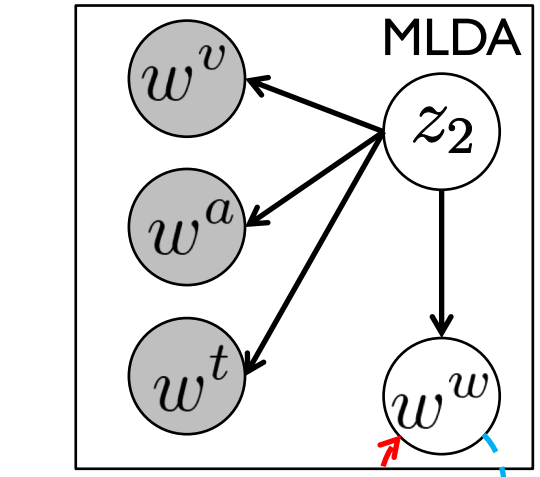
Example 4: Model for language acquisition

- Learn not only categories but also language model (LM)
  - Language model and object categories are learned mutually



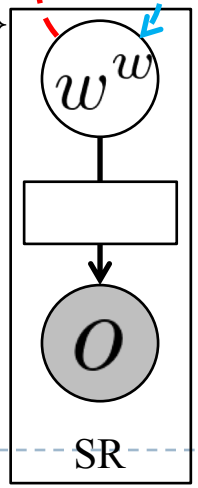
Example 4: Learning of LM

- SIR is used for learning language model



- Learn parameters of MLDA
- Compute $P(w^{(l)}|w^v, w^a, w^t)$ $(l = 1, \cdots, L)$

$\{w^{(1)}, \cdots, w^{(L)}\}$ $P(w^{(l)}|w^v, w^a, w^t)$



Samples speech recognition results

- w is determined by resampling based on $P(w^{(l)}|w^v, w^a, w^t)$
- Update parameters of LM

Example 4: Classification Results

- Classify 50 multimodal data

- Classification accuracy
  - w/o mutual learning: 80%
  - w/ mutual learning: 94%
- Speech recognition accuracy
  - w/o mutual learning: 64%
  - w/ mutual learning: 74%

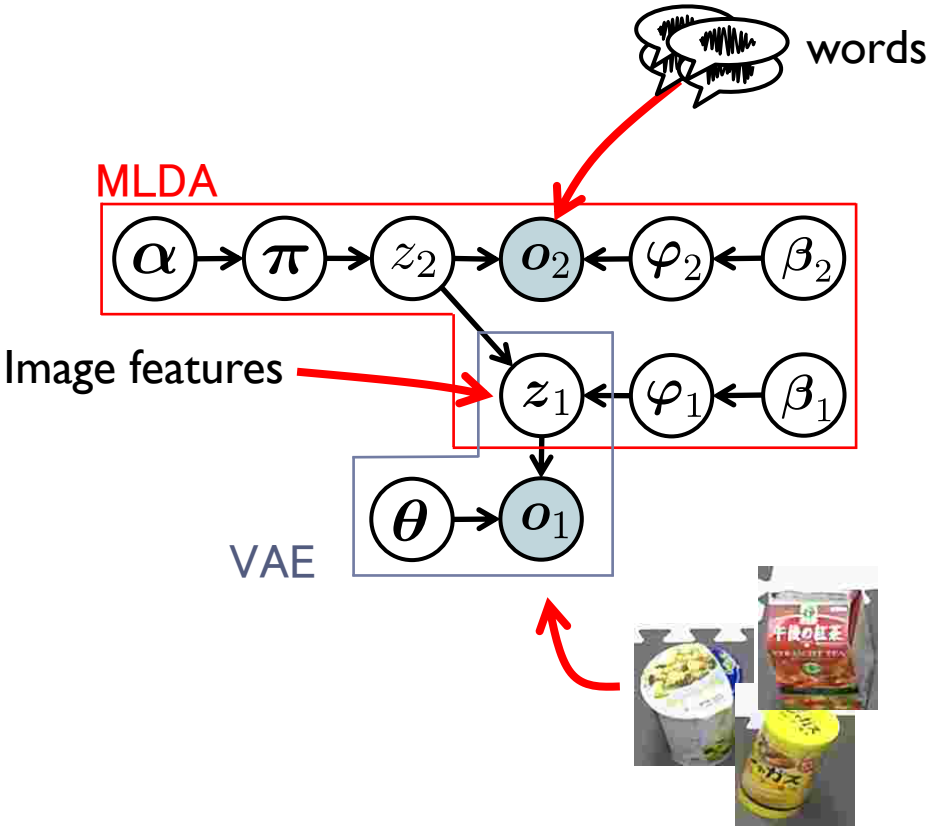
Classification and speech recognition accuracies improve by multimodal learning based on SERKET

Example 5: Model for learning object feature extractor by robots

- We used a dataset obtained through the robot observed objects and the human taught object feature by speech
  - # of the objects: 499
  - # of the categories: 81
- Words and images were used in this example
  - Words
    - The speech was recognized by phoneme recognizer
    - Recognized strings were segmented into words by unsupervised word segmentation (NPYLM)
  - Images
    - 
    - 
    - 

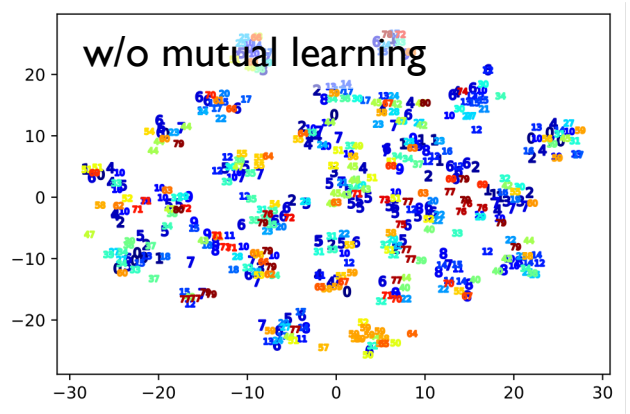
Example 5: MLDA+VAE

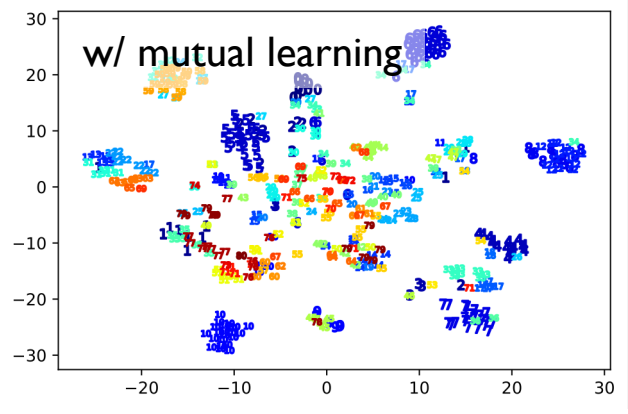
- The model to learn object categories and image features from words and images



Example 5: Classification Results

- Compare with the model where pre-trained CNN is used for feature extractor
- Classification accuracies:
  - Integrated model of VAE and MLDA: 67.8%
  - Model with pre-trained CNN: 66.8%
- Learned latent space:





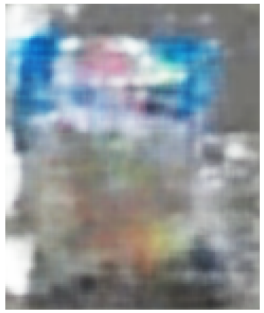
  - Suitable latent space for classification were learned

Example 5: Cross modal inference

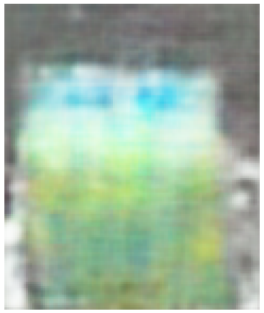
- Images were generated from words input

"Cup noodle"

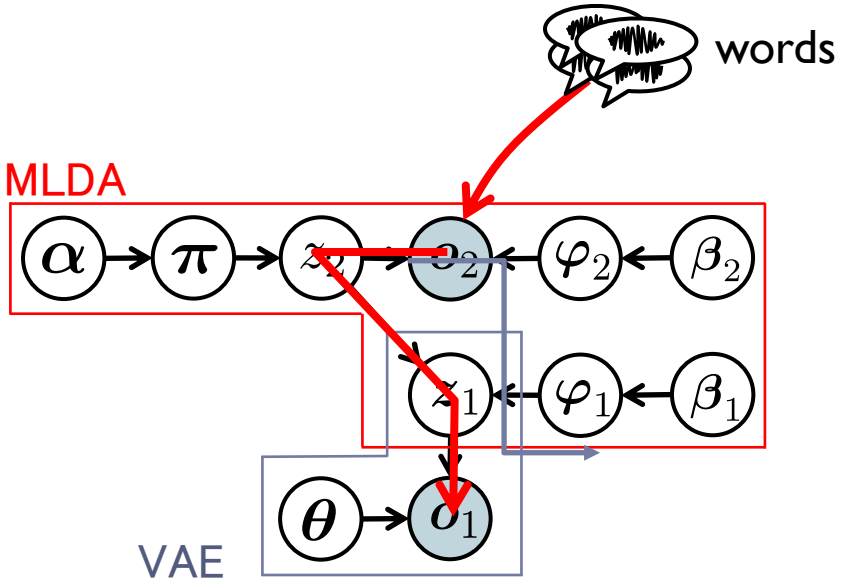
"Plastic bottle"



"Snack"



"Sponge"



- Images that represent the characteristics of the categories were generated

Conclusion

- We implemented VAE, GMM, LDA and MM modules, and integrated model with them
- We showed implementation examples and **it is easy to construct the integrated models**
- Accuracy improved by **mutual influence of modules**
- Moreover, we implemented speech recognition module and its integrated model

- Future work
  - Integrate deep neural network using Pixyz
  - Improve the efficiency of parameter inference