Adversarial Variational Embedding for Robust Semi-supervised Learning

J. Hao, M. Chen, W. Yu, Y. Sun and W. Wang

University of California, Los Angeles

KDD 2019

서울시립대학교
UNIVERSITY OF SEOUL

cheon.research@gmail.com

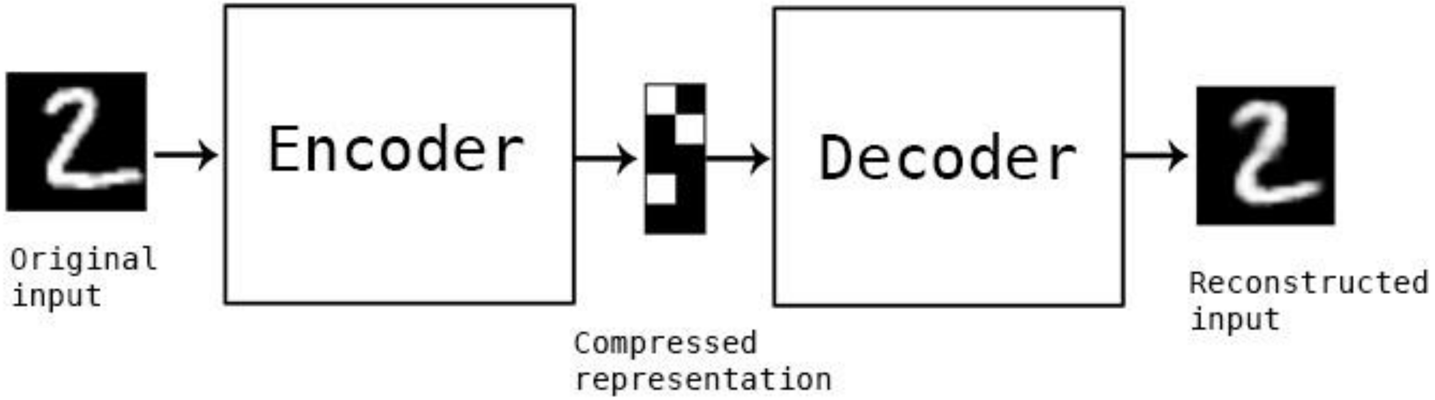
2020. 06. 02.

Keywords

Variational Autoencoder, Generative Adversarial Network, Representation Learning, Semi-Supervised Classification

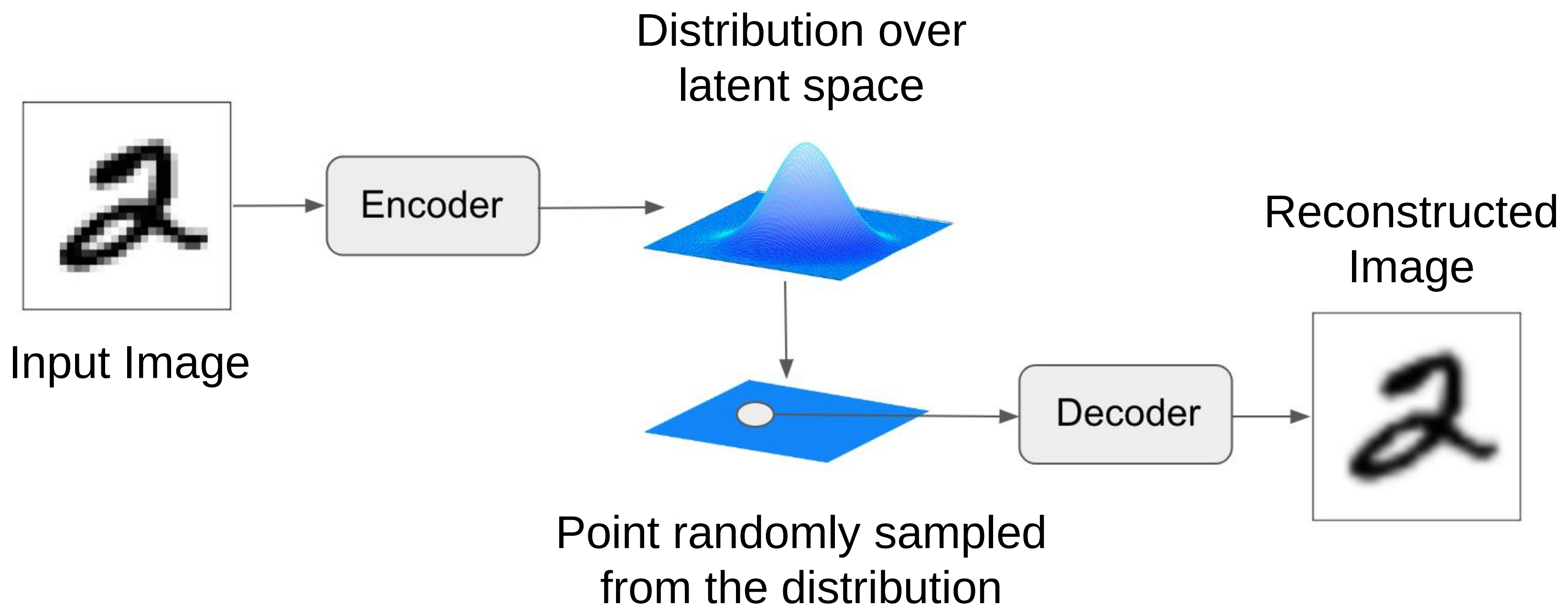
서울시립대학교 UNIVERSITY OF SEOUL

Autoencoders (AEs)

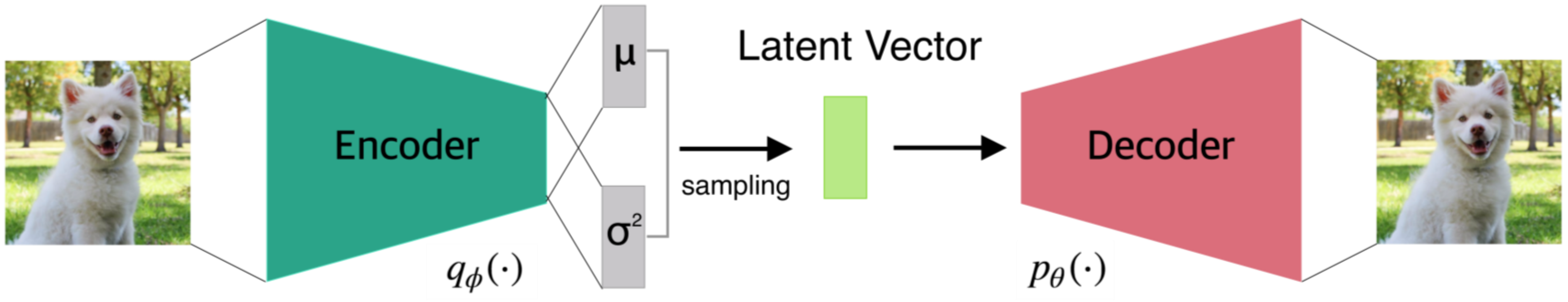


$$\min_{\phi,\theta} L_{rec} = \min \frac{1}{n} \sum_{i=1}^{n} \| \boldsymbol{x}_i - g_\theta(f_\phi(\boldsymbol{x}_i)) \|^2$$

Variational Autoencoders (VAEs)

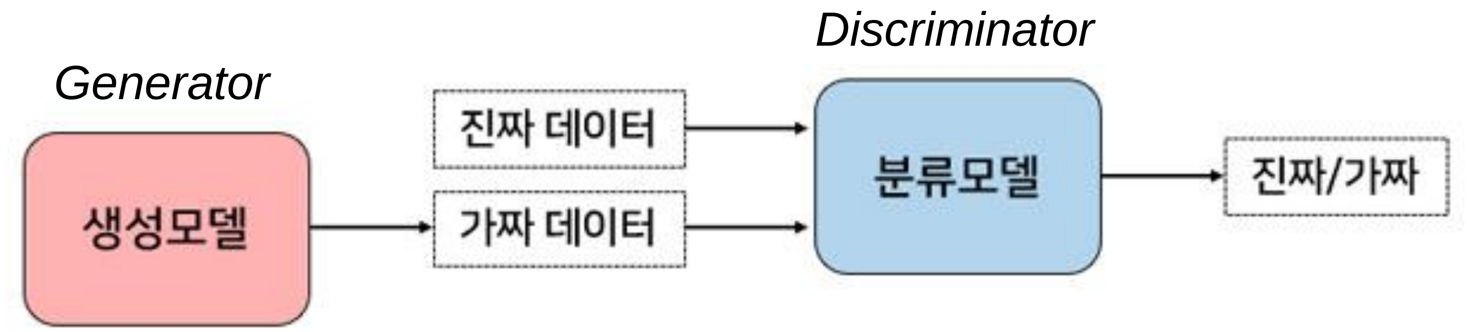

Distribution over latent space
Input Image
Encoder
Reconstructed Image
Decoder
Point randomly sampled from the distribution

Variational Autoencoders (VAEs)



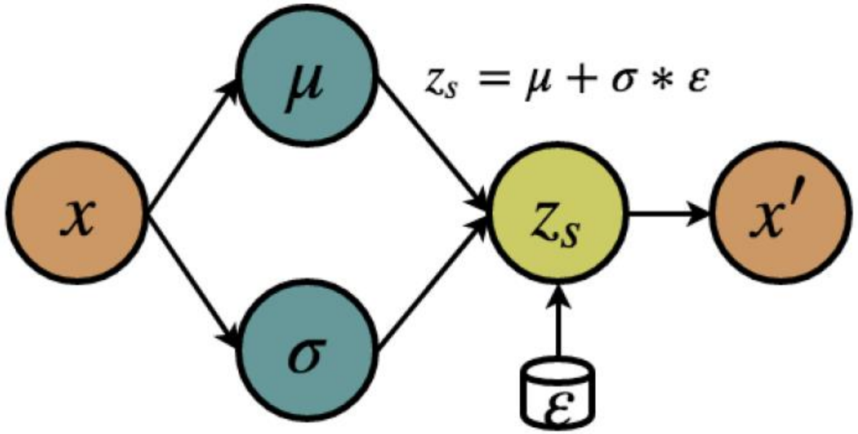
$$L = -E_{z\sim q(z|x)}\left[\log p(x|z)\right] + D_{KL}\left(q\left(z|x\right)||p\left(z\right)\right)$$

GAN (Generative Adversarial Network)



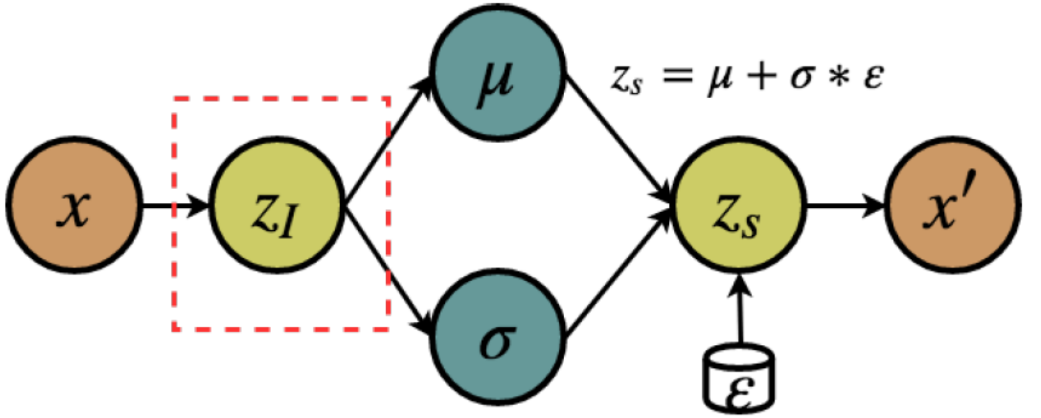
$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_{z}(z)}[\log(1 - D(G(z)))]$$

Motivation *- Why VAE++.*



(a) Standard VAE

Motivation *- Why VAE++.*



(b) VAE++

Motivation

- ❑ *Why VAE++ needs the Semi-supervised GAN.*

  To reduce the information loss between the two latent representations z_I and z_S to guarantee the learned z_I is representative.

- ❑ *Why Semi-supervised GAN needs the VAE++.*

  To provide a meaningful prior distribution that can represent the distribution of the input data.

Related Work

VAE for Semi-supervised Learning
Semi-supervised GAN
Combination of VAE and GAN

Proposed Method

1. To focus on **semi-supervised classification** instead of generation.

2. To attempt to learn an **exclusive latent representation** instead of a stochastic sampled representation.

3. To works on improvement of **latent space** instead of data space.

Methodology

Preliminary

Input dataset $(N = N_L + N_U)$

$\boldsymbol{x}_i^L \in \mathbb{R}^M \quad \boldsymbol{y}_i \in \mathbb{R}^K$

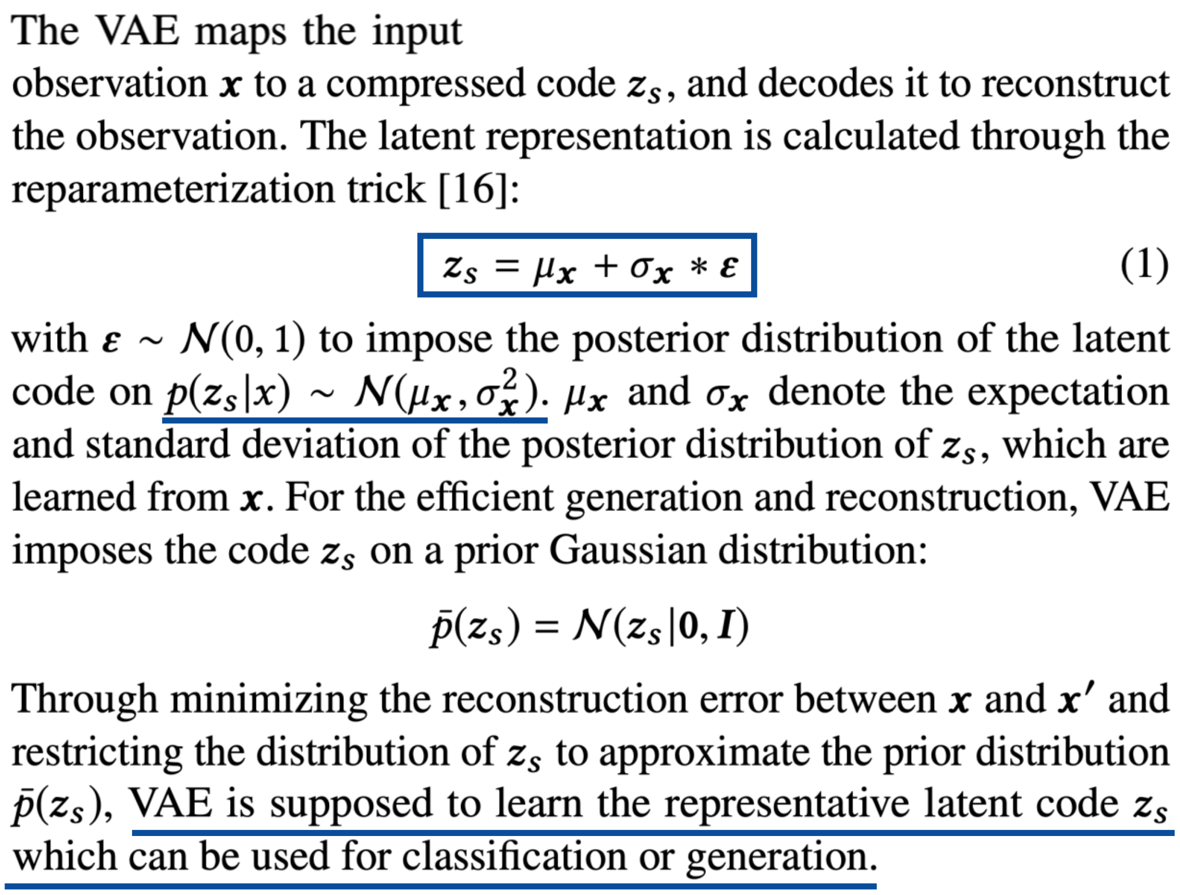
Labelled Samples $$(\boldsymbol{X}^L, \boldsymbol{Y}^L) = \{(\boldsymbol{x}_1^L, \boldsymbol{y}_1), (\boldsymbol{x}_2^L, \boldsymbol{y}_2), \cdots, (\boldsymbol{x}_{N_L}^L, \boldsymbol{y}_{N_L})\}$$

Unlabelled Samples $$\boldsymbol{X}^U = \{\boldsymbol{x}_1^U, \boldsymbol{x}_2^U, \cdots, \boldsymbol{x}_{N_U}^U\} \quad \boldsymbol{x}_i^U \in \mathbb{R}^M$$

1. To learn a **latent representation** which is rich of distinguishable information.

2. To build an encoder to provide an **embedding or feature representation** which allows accurate classification even with limited observations.

VAE



The VAE maps the input observation $\boldsymbol{x}$ to a compressed code $\boldsymbol{z_s}$, and decodes it to reconstruct the observation. The latent representation is calculated through the reparameterization trick [16]:

$$\boxed{\boldsymbol{z_s} = \mu_{\boldsymbol{x}} + \sigma_{\boldsymbol{x}} * \boldsymbol{\varepsilon}} \tag{1}$$

with $\boldsymbol{\varepsilon} \sim \mathcal{N}(0,1)$ to impose the posterior distribution of the latent code on $\underline{p(\boldsymbol{z_s}|x) \sim \mathcal{N}(\mu_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}}^2)}$. $\mu_{\boldsymbol{x}}$ and $\sigma_{\boldsymbol{x}}$ denote the expectation and standard deviation of the posterior distribution of $\boldsymbol{z_s}$, which are learned from $\boldsymbol{x}$. For the efficient generation and reconstruction, VAE imposes the code $\boldsymbol{z_s}$ on a prior Gaussian distribution:

$$\bar{p}(\boldsymbol{z_s}) = \mathcal{N}(\boldsymbol{z_s}|\boldsymbol{0}, \boldsymbol{I})$$

Through minimizing the reconstruction error between $\boldsymbol{x}$ and $\boldsymbol{x}'$ and restricting the distribution of $\boldsymbol{z_s}$ to approximate the prior distribution $\bar{p}(\boldsymbol{z_s})$, VAE is supposed to learn the representative latent code $\boldsymbol{z_s}$ which can be used for classification or generation.

Limitation of Standard VAE

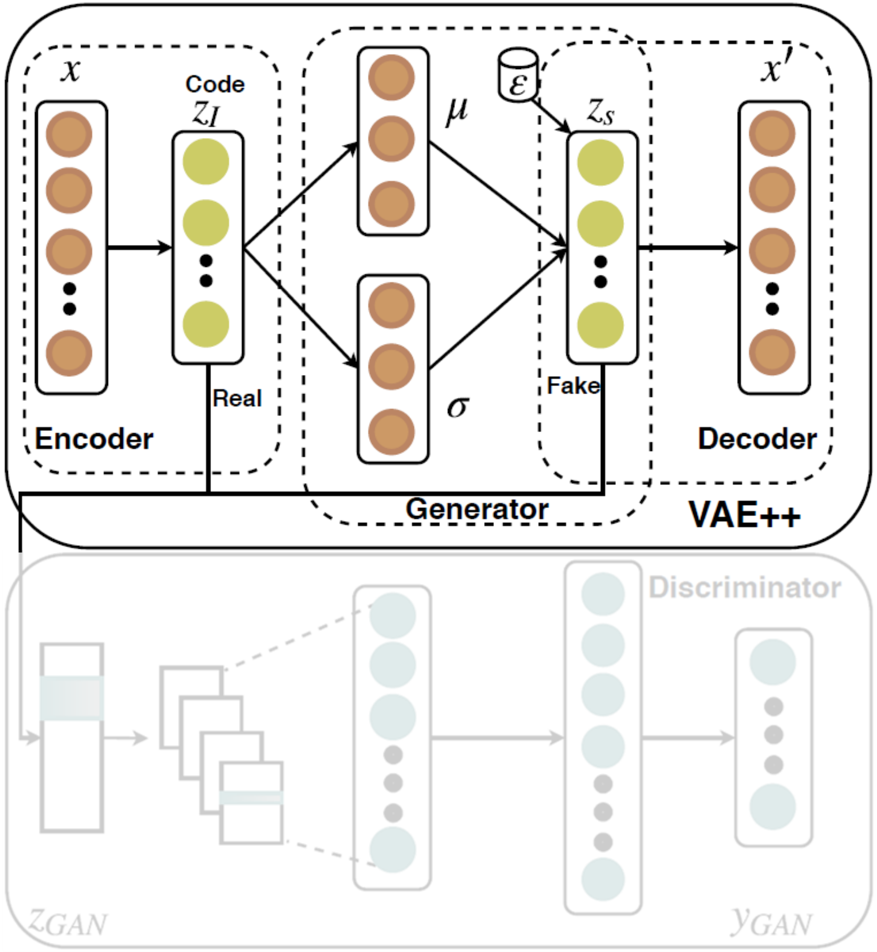
$$z_s = g(\mu_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}}, \boldsymbol{\varepsilon})$$

The learned latent code is not **exclusive**.

The latent code $\boldsymbol{z}_s$ is determined by two factors.

- the prior distribution of observation $\bar{p}(\boldsymbol{x})$
- the stochastically sampled data $\boldsymbol{\varepsilon}$

AVAE
- VAE++



x
Code
z_I
μ
ε
z_s
x'
Real
σ
Fake
Encoder
Decoder
Generator
VAE++
Discriminator
z_{GAN}
y_{GAN}

VAE++ – Encoder and Generator

non-linear transformation

$$p_{\theta_{en}}(z_I|\boldsymbol{x}) = f(z_I; \boldsymbol{x}, \theta_{en})$$

encoder parameters

exclusive latent code $z_I \in \mathbb{R}^D$

$$z_S = \mu(z_I) + \sigma(z_I) * \boldsymbol{\varepsilon}$$

VAE++ – Decoder and Loss

$$\underset{\substack{\downarrow \\ \textit{decoder parameters}}}{p_{\theta_{de}}}(\boldsymbol{x}'|z_s) = \overset{\substack{\textit{non-linear rendering} \\ \uparrow}}{f'}(\underset{\substack{\downarrow \\ \textit{reconstructed observation}}}{\boldsymbol{x}'}; z_s, \theta_{de})$$

$$\mathcal{L}_{VAE} = -\mathbb{E}_{z_s \sim p_{\theta_{en}}(z_s|\boldsymbol{x})}[\log p_{\theta_{de}}(\boldsymbol{x}'|z_s)] + KL(p_{\theta_{en}}(z_s|\boldsymbol{x})||\bar{p}(z_s))$$

VAE++

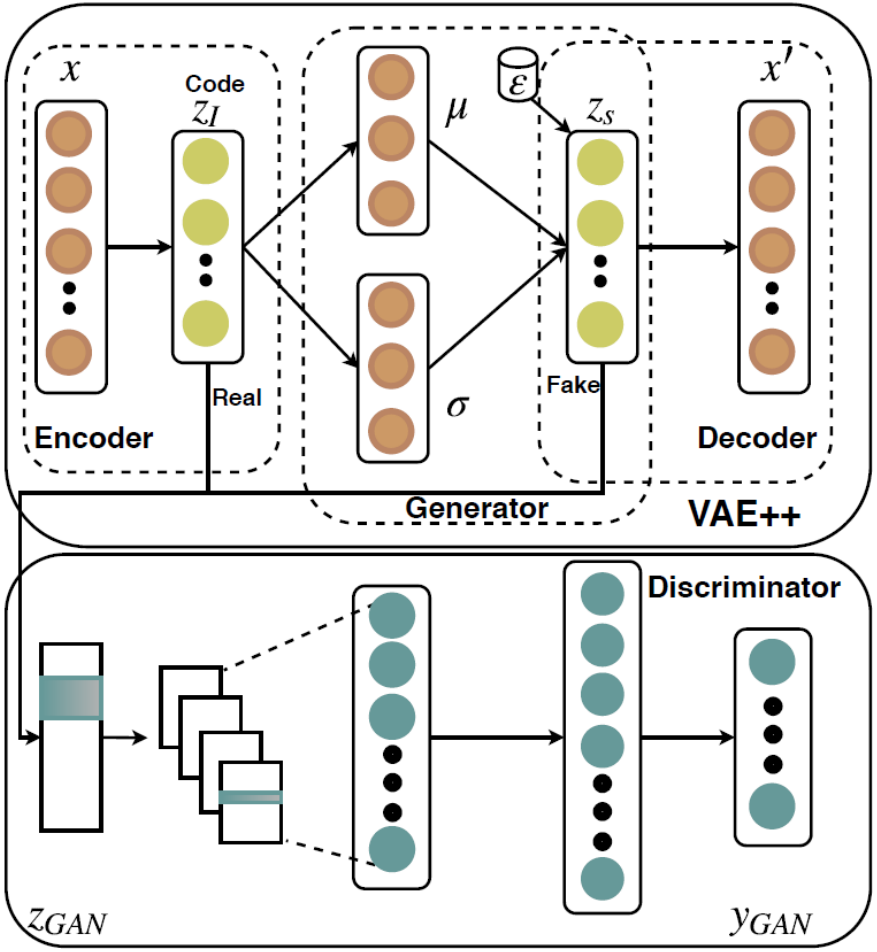
$$z_s = \mu(z_I) + \sigma(z_I) * \boldsymbol{\varepsilon}$$

From equation (2), we can observe that the expectation and standard deviation of $\boldsymbol{z}_s$ and $\boldsymbol{z}_I$ are invariant. In particular, for a specific sample $\boldsymbol{x}_i$, the corresponding $\boldsymbol{z}_{si}$ and $\boldsymbol{z}_{Ii}$ have the same statistical characteristics. Thus, we have

$$\boldsymbol{z}_s \leftarrow \mu(\boldsymbol{z}_I), \sigma(\boldsymbol{z}_I), \boldsymbol{\varepsilon}$$

which indicates that the generated $\boldsymbol{z}_s$ is affected by both the distribution (or statistic characteristics) of $\boldsymbol{z}_I$ and the prior distribution $\bar{p}(\boldsymbol{z}_s)$ (or $\boldsymbol{\varepsilon}$). In summary, the $\boldsymbol{z}_s$ inherits the statistical characteristics of $\boldsymbol{z}_I$.

AVAE



x
Code
z_I
μ
ε
z_s
x'
Real
σ
Fake
Encoder
Decoder
Generator
VAE++
Discriminator
z_{GAN}
y_{GAN}

Adversarial Variational Embedding

❑ Generator

For labelled observations $\boldsymbol{x}^L \rightarrow \boldsymbol{z}_I^L \in \mathbb{R}^D, \ \boldsymbol{z}_s^L \in \mathbb{R}^D$

For unlabelled observations $\boldsymbol{x}^U \rightarrow \boldsymbol{z}_I^U \in \mathbb{R}^D, \ \boldsymbol{z}_s^U \in \mathbb{R}^D$

$$\boldsymbol{y} \in \mathbb{R}^K \Rightarrow \boldsymbol{y}_{GAN} \in \mathbb{R}^{K+1}$$

K possible classes to *K* + *1* possible classes regarding generated fake samples

Adversarial Variational Embedding

- Discriminator

input of the discriminator

$$q_{\varphi}(y_{GAN} | z_{GAN}) = h(y_{GAN}; z_{GAN}, \varphi)$$

discriminator parameters

non-linear transformation

z_{GAN} is fake (from z_S) $\quad q_{\varphi}(y_{GAN} = K + 1 | z_{GAN})$

z_{GAN} is real (from z_I) $\quad q_{\varphi}(y_{GAN} | z_{GAN}, y_{GAN} < K + 1)$

Adversarial Variational Embedding

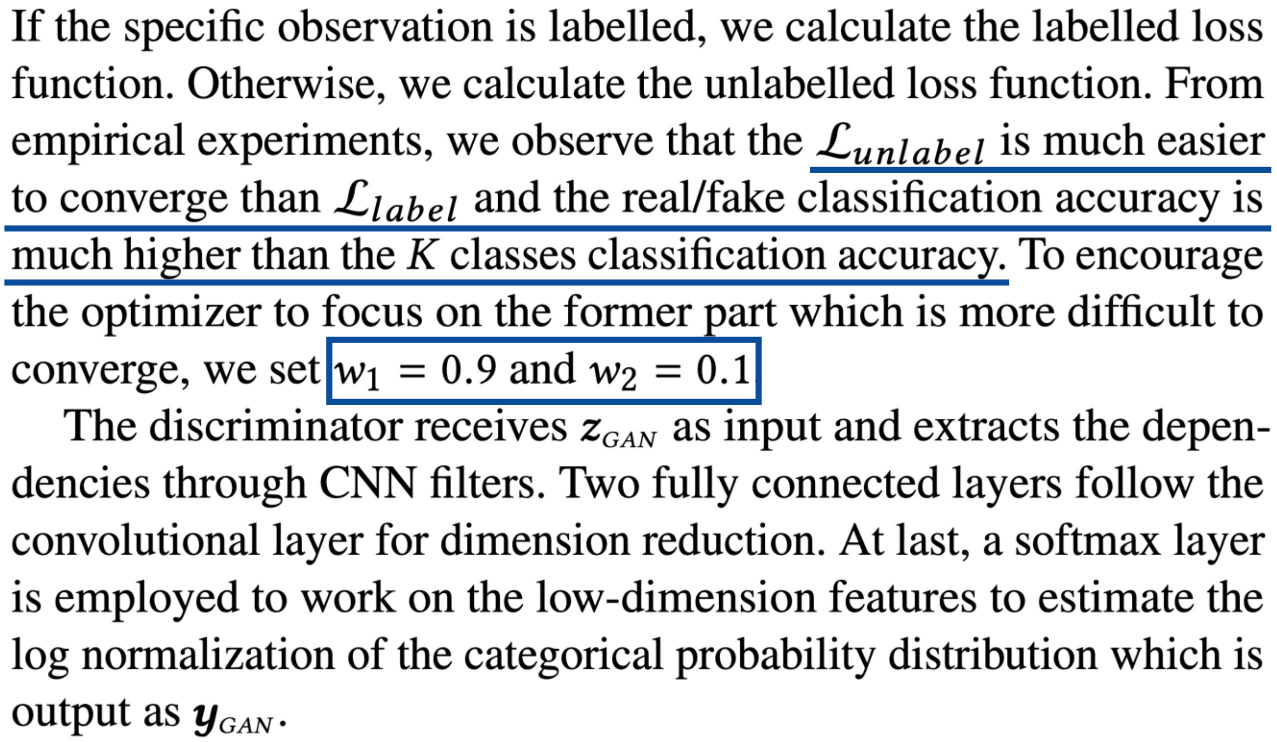
$$\mathcal{L}_{label} = -\mathbb{E}_{\boldsymbol{z}_{GAN},\boldsymbol{y}_{GAN}\sim p_j}[log\boldsymbol{q}_{\boldsymbol{\varphi}}(\boldsymbol{y}_{\mathrm{GAN}}|\boldsymbol{z}_{\mathrm{GAN}},\boldsymbol{y}_{\mathrm{GAN}} < K+1)]$$

$$\begin{aligned}\mathcal{L}_{unlabel} &= -\mathbb{E}_{\boldsymbol{z}_{GAN}\sim p_{\boldsymbol{\theta}_{en}}(\boldsymbol{z}_I|\boldsymbol{x})}[log(1-\boldsymbol{q}_{\boldsymbol{\varphi}}(\boldsymbol{y}_{\mathrm{GAN}} = K+1|\boldsymbol{z}_{\mathrm{GAN}}))] \\ &\quad - \mathbb{E}_{\boldsymbol{z}_{GAN}\sim p_{\boldsymbol{\theta}_{en}}(\boldsymbol{z}_s|\boldsymbol{x})}[log(\boldsymbol{q}_{\boldsymbol{\varphi}}(\boldsymbol{y}_{\mathrm{GAN}} = K+1|\boldsymbol{z}_{\mathrm{GAN}}))]\end{aligned}$$

$$\mathcal{L}_{GAN} = w_1 * flag * \mathcal{L}_{label} + w_2 * (1 - flag) * \mathcal{L}_{unlabel}$$

$w_1,\ w_2$: weight $\quad flag = \begin{cases} 1 & labelled \\ 0 & unlabelled \end{cases}$

Adversarial Variational Embedding

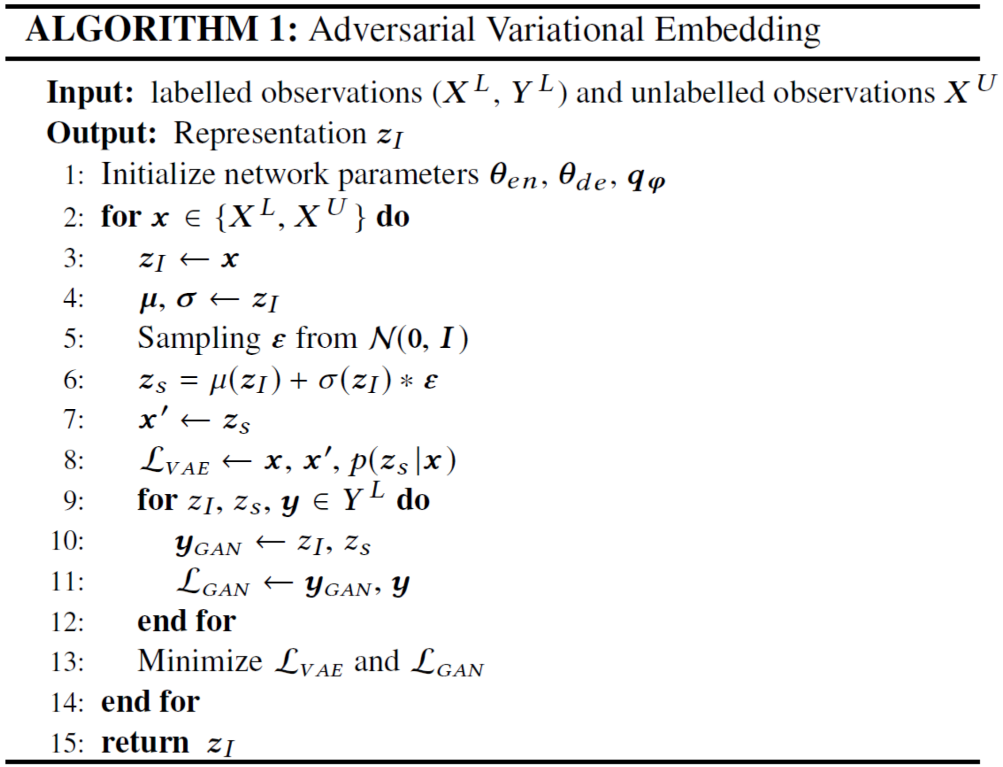


If the specific observation is labelled, we calculate the labelled loss function. Otherwise, we calculate the unlabelled loss function. From empirical experiments, we observe that the $\mathcal{L}_{unlabel}$ is much easier to converge than $\mathcal{L}_{label}$ and the real/fake classification accuracy is much higher than the K classes classification accuracy. To encourage the optimizer to focus on the former part which is more difficult to converge, we set $\boxed{w_1 = 0.9 \text{ and } w_2 = 0.1}$

The discriminator receives $\boldsymbol{z}_{GAN}$ as input and extracts the dependencies through CNN filters. Two fully connected layers follow the convolutional layer for dimension reduction. At last, a softmax layer is employed to work on the low-dimension features to estimate the log normalization of the categorical probability distribution which is output as $\boldsymbol{y}_{GAN}$.

서울시립대학교
UNIVERSITY OF SEOUL

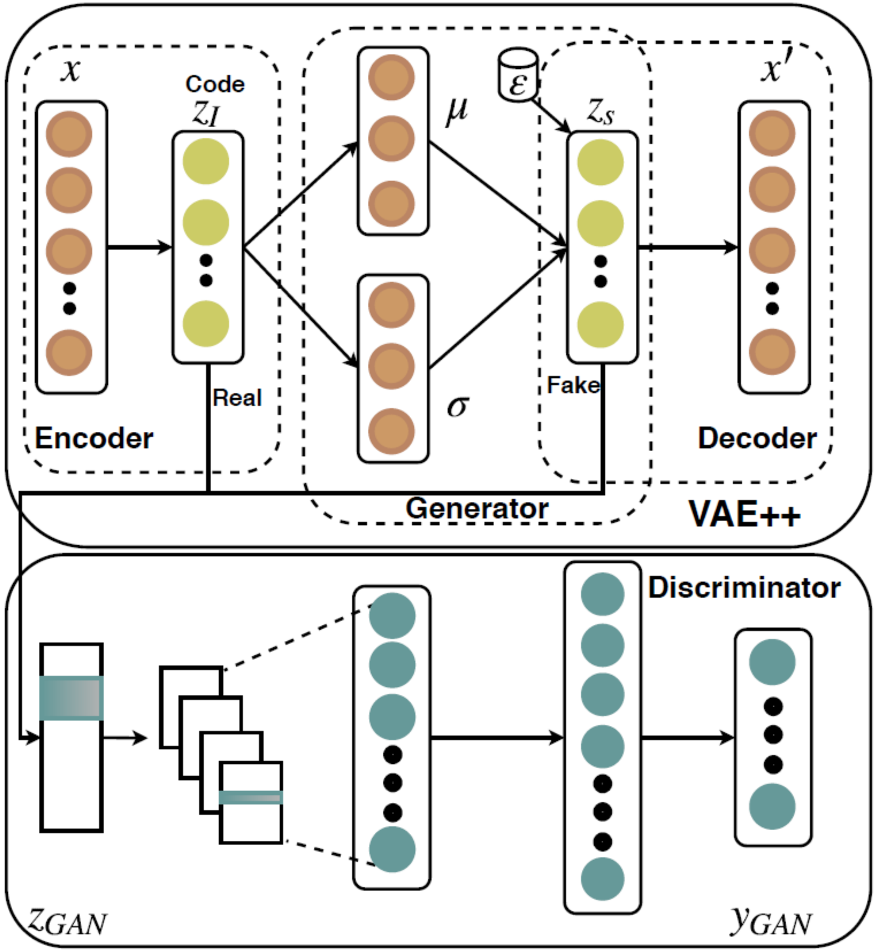
Adversarial Variational Embedding



ALGORITHM 1: Adversarial Variational Embedding

```
Input: labelled observations ($X^L$, $Y^L$) and unlabelled observations $X^U$
Output: Representation $z_I$
 1: Initialize network parameters $\theta_{en}$, $\theta_{de}$, $q_\varphi$
 2: for $x \in \{X^L, X^U\}$ do
 3:     $z_I \leftarrow x$
 4:     $\mu, \sigma \leftarrow z_I$
 5:     Sampling $\varepsilon$ from $\mathcal{N}(0, I)$
 6:     $z_s = \mu(z_I) + \sigma(z_I) * \varepsilon$
 7:     $x' \leftarrow z_s$
 8:     $\mathcal{L}_{VAE} \leftarrow x, x', p(z_s|x)$
 9:     for $z_I, z_s, y \in Y^L$ do
10:         $y_{GAN} \leftarrow z_I, z_s$
11:         $\mathcal{L}_{GAN} \leftarrow y_{GAN}, y$
12:     end for
13:     Minimize $\mathcal{L}_{VAE}$ and $\mathcal{L}_{GAN}$
14: end for
15: return $z_I$
```

AVAE


x
Code
z_I
μ
ε
z_s
x'
Real
σ
Fake
Encoder
Decoder
Generator
VAE++
Discriminator
z_{GAN}
y_{GAN}

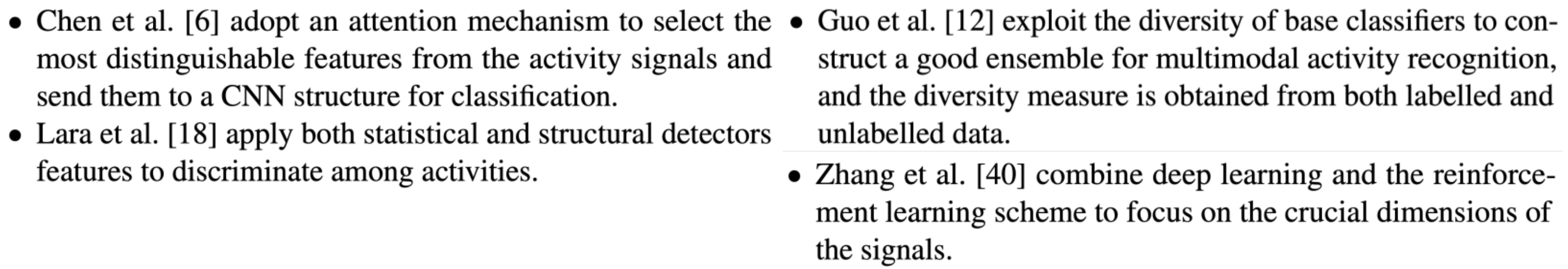
Experiments

Experiments – (1) Activity Recognition

❑ Experiment Setup

- PAMAP2 dataset - Cycling, standing, walking, lying and running.
- The time window is set as 10 with 50% overlapping.
- Training set 80%, testing set 20%

❑ Application-related State-of-the-arts



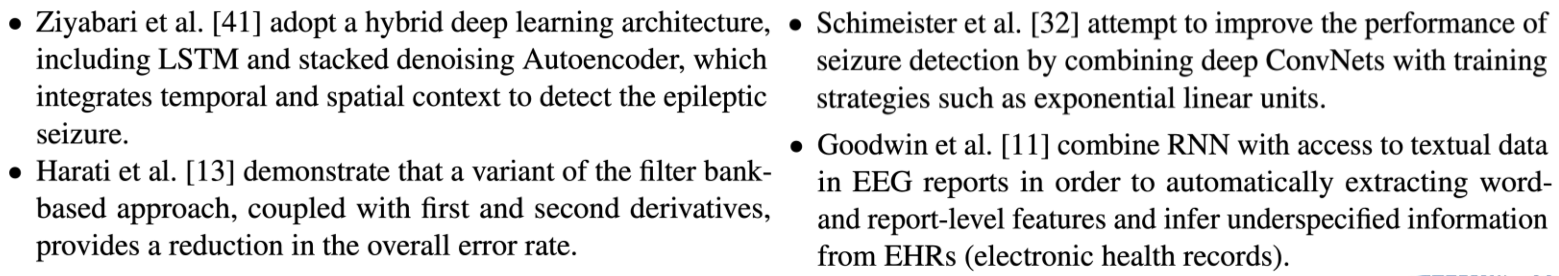
- Chen et al. [6] adopt an attention mechanism to select the most distinguishable features from the activity signals and send them to a CNN structure for classification.
- Lara et al. [18] apply both statistical and structural detectors features to discriminate among activities.
- Guo et al. [12] exploit the diversity of base classifiers to construct a good ensemble for multimodal activity recognition, and the diversity measure is obtained from both labelled and unlabelled data.
- Zhang et al. [40] combine deep learning and the reinforcement learning scheme to focus on the crucial dimensions of the signals.

Experiments – (2) Neurological Diagnosis

❑ Experiment Setup

- TUH dataset – EEG data (Epileptic seizure state and normal state)
- The sampling rate is set as 250Hz
- Select 12,000 samples from each of 18 subjects.

❑ Application-related State-of-the-arts

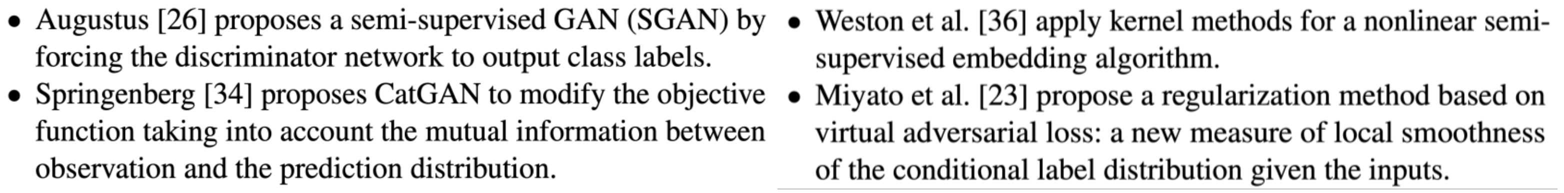


- Ziyabari et al. [41] adopt a hybrid deep learning architecture, including LSTM and stacked denoising Autoencoder, which integrates temporal and spatial context to detect the epileptic seizure.
- Harati et al. [13] demonstrate that a variant of the filter bank-based approach, coupled with first and second derivatives, provides a reduction in the overall error rate.
- Schimeister et al. [32] attempt to improve the performance of seizure detection by combining deep ConvNets with training strategies such as exponential linear units.
- Goodwin et al. [11] combine RNN with access to textual data in EEG reports in order to automatically extracting word- and report-level features and infer underspecified information from EHRs (electronic health records).

Experiments – (3) Image Classification

- Experiment Setup
  - To evaluate the representation learning ability in images.
  - MNIST dataset – Handwritten(0-9) digital images.
  - 50,000 for training, 10,000 for testing.

- Application-related State-of-the-arts



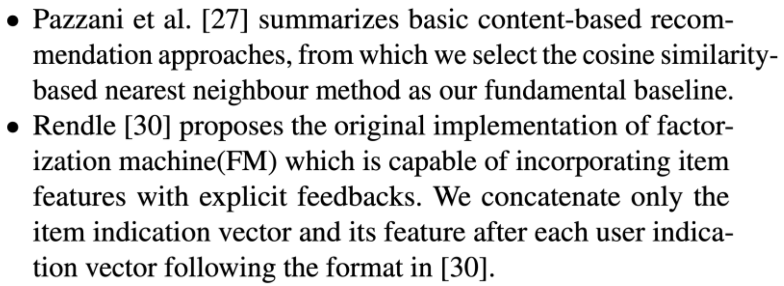
- Augustus [26] proposes a semi-supervised GAN (SGAN) by forcing the discriminator network to output class labels.
- Springenberg [34] proposes CatGAN to modify the objective function taking into account the mutual information between observation and the prediction distribution.
- Weston et al. [36] apply kernel methods for a nonlinear semi-supervised embedding algorithm.
- Miyato et al. [23] propose a regularization method based on virtual adversarial loss: a new measure of local smoothness of the conditional label distribution given the inputs.

Experiments – (4) Recommender System

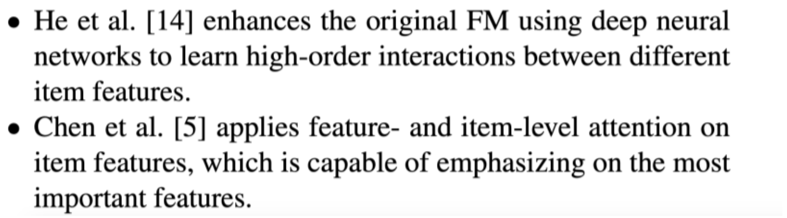
❑ Experiment Setup

- Yelp dataset – Restaurant attributes(13) and ratings(1-5)
- A unseen business's attributes as input data and predict the possible ratings from the potential customers.

❑ Application-related State-of-the-arts

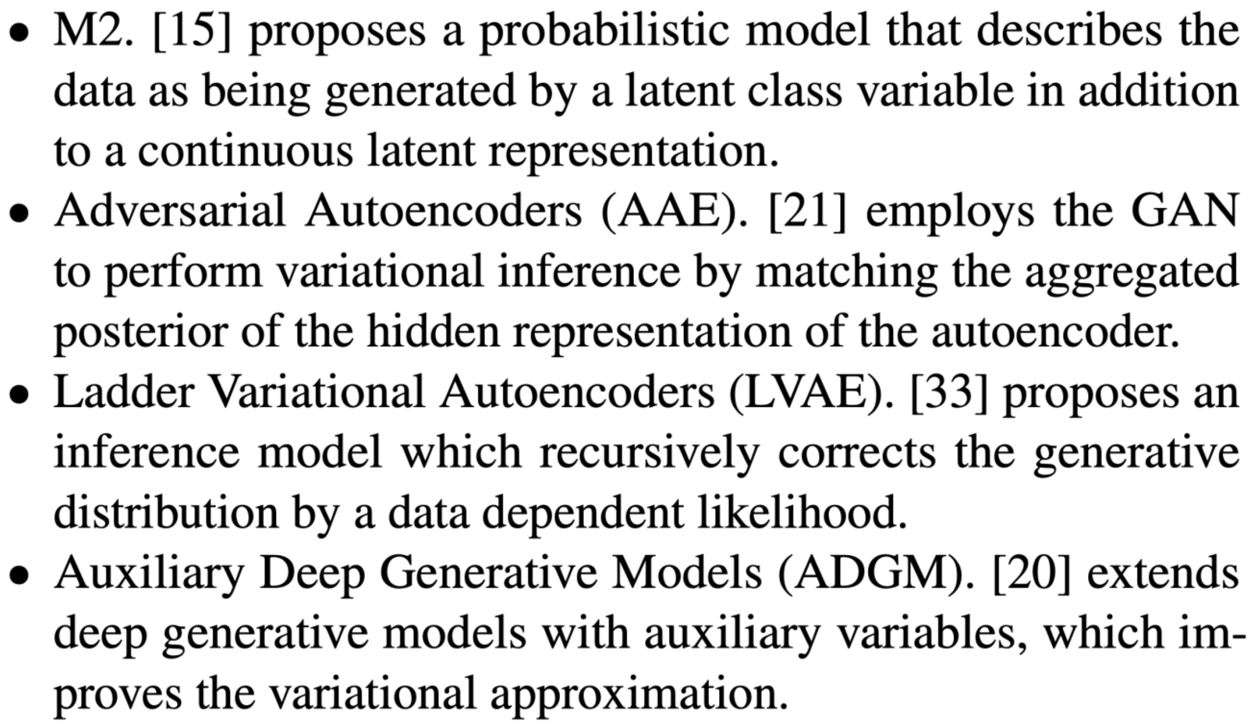


- Pazzani et al. [27] summarizes basic content-based recommendation approaches, from which we select the cosine similarity-based nearest neighbour method as our fundamental baseline.
- Rendle [30] proposes the original implementation of factorization machine(FM) which is capable of incorporating item features with explicit feedbacks. We concatenate only the item indication vector and its feature after each user indication vector following the format in [30].



- He et al. [14] enhances the original FM using deep neural networks to learn high-order interactions between different item features.
- Chen et al. [5] applies feature- and item-level attention on item features, which is capable of emphasizing on the most important features.

Experiments – Algorithm-related SOTA



- M2. [15] proposes a probabilistic model that describes the data as being generated by a latent class variable in addition to a continuous latent representation.
- Adversarial Autoencoders (AAE). [21] employs the GAN to perform variational inference by matching the aggregated posterior of the hidden representation of the autoencoder.
- Ladder Variational Autoencoders (LVAE). [33] proposes an inference model which recursively corrects the generative distribution by a data dependent likelihood.
- Auxiliary Deep Generative Models (ADGM). [20] extends deep generative models with auxiliary variables, which improves the variational approximation.

Semi-supervised Classification Accuracy(1)

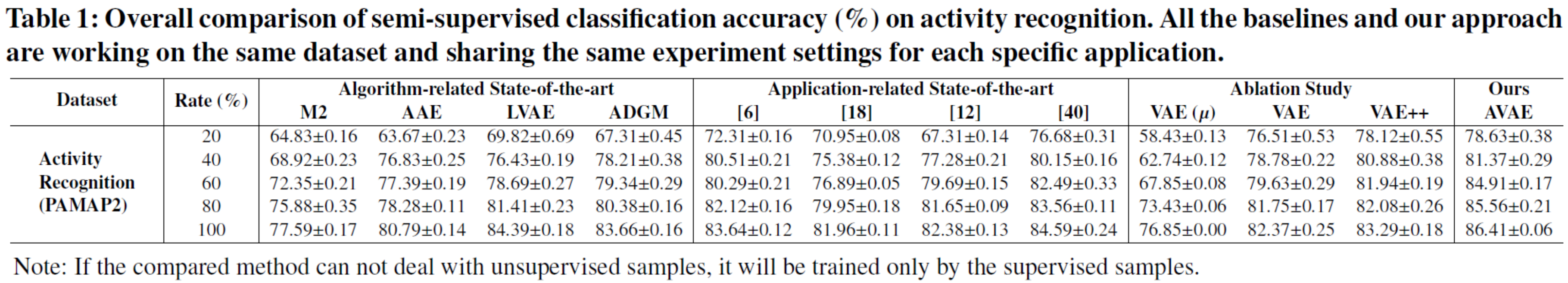


Table 1: Overall comparison of semi-supervised classification accuracy (%) on activity recognition. All the baselines and our approach are working on the same dataset and sharing the same experiment settings for each specific application.

Dataset	Rate (%)	Algorithm-related State-of-the-art				Application-related State-of-the-art				Ablation Study			Ours
		M2	AAE	LVAE	ADGM	[6]	[18]	[12]	[40]	VAE (μ)	VAE	VAE++	AVAE
Activity Recognition (PAMAP2)	20	64.83±0.16	63.67±0.23	69.82±0.69	67.31±0.45	72.31±0.16	70.95±0.08	67.31±0.14	76.68±0.31	58.43±0.13	76.51±0.53	78.12±0.55	78.63±0.38
	40	68.92±0.23	76.83±0.25	76.43±0.19	78.21±0.38	80.51±0.21	75.38±0.12	77.28±0.21	80.15±0.16	62.74±0.12	78.78±0.22	80.88±0.38	81.37±0.29
	60	72.35±0.21	77.39±0.19	78.69±0.27	79.34±0.29	80.29±0.21	76.89±0.05	79.69±0.15	82.49±0.33	67.85±0.08	79.63±0.29	81.94±0.19	84.91±0.17
	80	75.88±0.35	78.28±0.11	81.41±0.23	80.38±0.16	82.12±0.16	79.95±0.18	81.65±0.09	83.56±0.11	73.43±0.06	81.75±0.17	82.08±0.26	85.56±0.21
	100	77.59±0.17	80.79±0.14	84.39±0.18	83.66±0.16	83.64±0.12	81.96±0.11	82.38±0.13	84.59±0.24	76.85±0.00	82.37±0.25	83.29±0.18	86.41±0.06

Note: If the compared method can not deal with unsupervised samples, it will be trained only by the supervised samples.

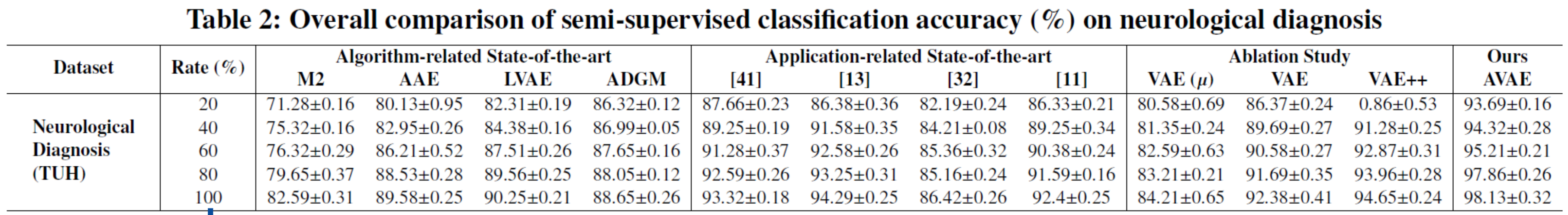


Table 2: Overall comparison of semi-supervised classification accuracy (%) on neurological diagnosis

Dataset	Rate (%)	Algorithm-related State-of-the-art				Application-related State-of-the-art				Ablation Study			Ours
		M2	AAE	LVAE	ADGM	[41]	[13]	[32]	[11]	VAE (μ)	VAE	VAE++	AVAE
Neurological Diagnosis (TUH)	20	71.28±0.16	80.13±0.95	82.31±0.19	86.32±0.12	87.66±0.23	86.38±0.36	82.19±0.24	86.33±0.21	80.58±0.69	86.37±0.24	0.86±0.53	93.69±0.16
	40	75.32±0.16	82.95±0.26	84.38±0.16	86.99±0.05	89.25±0.19	91.58±0.35	84.21±0.08	89.25±0.34	81.35±0.24	89.69±0.27	91.28±0.25	94.32±0.28
	60	76.32±0.29	86.21±0.52	87.51±0.26	87.65±0.16	91.28±0.37	92.58±0.26	85.36±0.32	90.38±0.24	82.59±0.63	90.58±0.27	92.87±0.31	95.21±0.21
	80	79.65±0.37	88.53±0.28	89.56±0.25	88.05±0.12	92.59±0.26	93.25±0.31	85.16±0.24	91.59±0.16	83.21±0.21	91.69±0.35	93.96±0.28	97.86±0.26
	100	82.59±0.31	89.58±0.25	90.25±0.21	88.65±0.26	93.32±0.18	94.29±0.25	86.42±0.26	92.4±0.25	84.21±0.65	92.38±0.41	94.65±0.24	98.13±0.32

supervision rate

$$N^L/(N^L + N^U)$$

Semi-supervised Classification Accuracy(2)

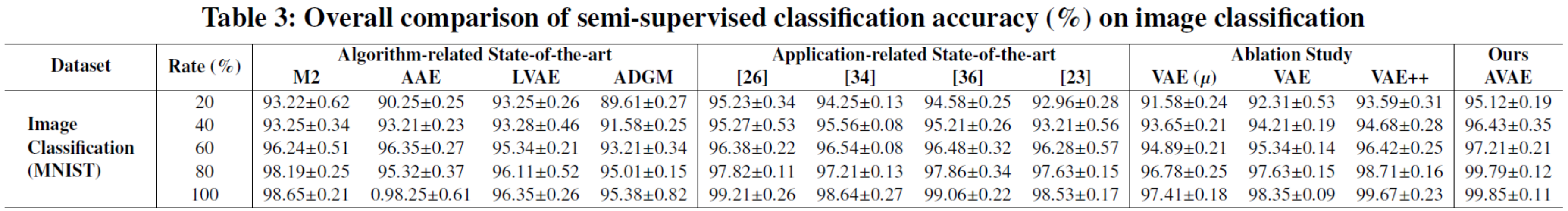


Table 3: Overall comparison of semi-supervised classification accuracy (%) on image classification

Dataset	Rate (%)	Algorithm-related State-of-the-art				Application-related State-of-the-art				Ablation Study			Ours
		M2	AAE	LVAE	ADGM	[26]	[34]	[36]	[23]	VAE (μ)	VAE	VAE++	AVAE
Image Classification (MNIST)	20	93.22±0.62	90.25±0.25	93.25±0.26	89.61±0.27	95.23±0.34	94.25±0.13	94.58±0.25	92.96±0.28	91.58±0.24	92.31±0.53	93.59±0.31	95.12±0.19
	40	93.25±0.34	93.21±0.23	93.28±0.46	91.58±0.25	95.27±0.53	95.56±0.08	95.21±0.26	93.21±0.56	93.65±0.21	94.21±0.19	94.68±0.28	96.43±0.35
	60	96.24±0.51	96.35±0.27	95.34±0.21	93.21±0.34	96.38±0.22	96.54±0.08	96.48±0.32	96.28±0.57	94.89±0.21	95.34±0.14	96.42±0.25	97.21±0.21
	80	98.19±0.25	95.32±0.37	96.11±0.52	95.01±0.15	97.82±0.11	97.21±0.13	97.86±0.34	97.63±0.15	96.78±0.25	97.63±0.15	98.71±0.16	99.79±0.12
	100	98.65±0.21	0.98.25±0.61	96.35±0.26	95.38±0.82	99.21±0.26	98.64±0.27	99.06±0.22	98.53±0.17	97.41±0.18	98.35±0.09	99.67±0.23	99.85±0.11

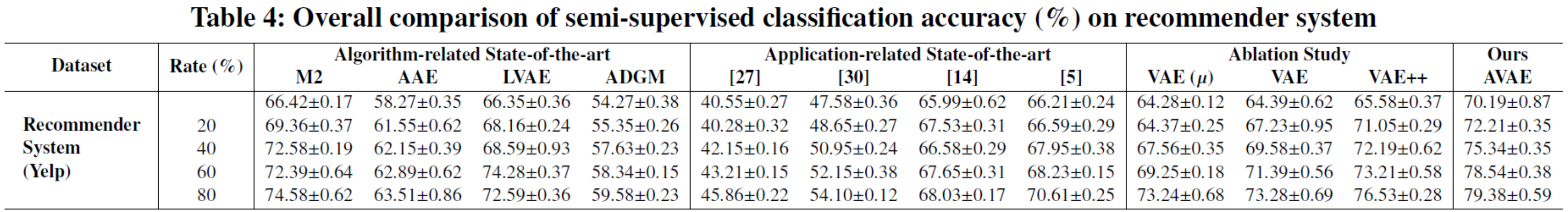
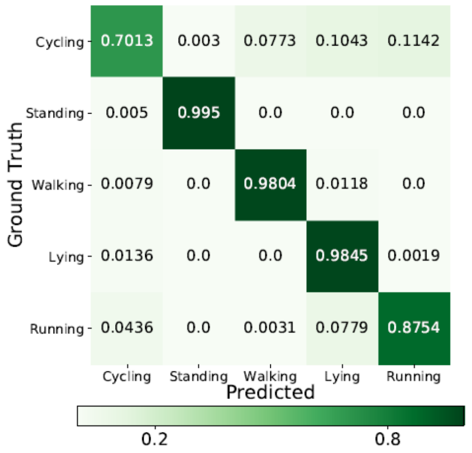


Table 4: Overall comparison of semi-supervised classification accuracy (%) on recommender system

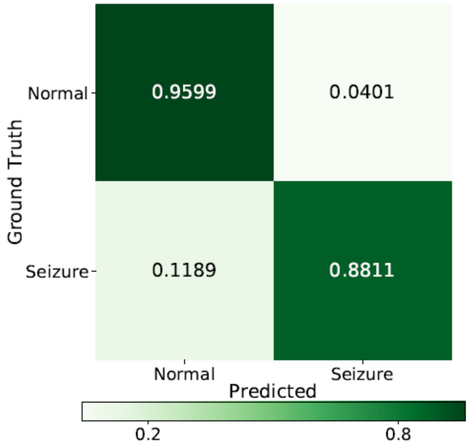
Dataset	Rate (%)	Algorithm-related State-of-the-art				Application-related State-of-the-art				Ablation Study			Ours
		M2	AAE	LVAE	ADGM	[27]	[30]	[14]	[5]	VAE (μ)	VAE	VAE++	AVAE
Recommender System (Yelp)		66.42±0.17	58.27±0.35	66.35±0.36	54.27±0.38	40.55±0.27	47.58±0.36	65.99±0.62	66.21±0.24	64.28±0.12	64.39±0.62	65.58±0.37	70.19±0.87
	20	69.36±0.37	61.55±0.62	68.16±0.24	55.35±0.26	40.28±0.32	48.65±0.27	67.53±0.31	66.59±0.29	64.37±0.25	67.23±0.95	71.05±0.29	72.21±0.35
	40	72.58±0.19	62.15±0.39	68.59±0.93	57.63±0.23	42.15±0.16	50.95±0.24	66.58±0.29	67.95±0.38	67.56±0.35	69.58±0.37	72.19±0.62	75.34±0.35
	60	72.39±0.64	62.89±0.62	74.28±0.37	58.34±0.15	43.21±0.15	52.15±0.38	67.65±0.31	68.23±0.15	69.25±0.18	71.39±0.56	73.21±0.58	78.54±0.38
	80	74.58±0.62	63.51±0.86	72.59±0.36	59.58±0.23	45.86±0.22	54.10±0.12	68.03±0.17	70.61±0.25	73.24±0.68	73.28±0.69	76.53±0.28	79.38±0.59

서울시립대학교
UNIVERSITY OF SEOUL

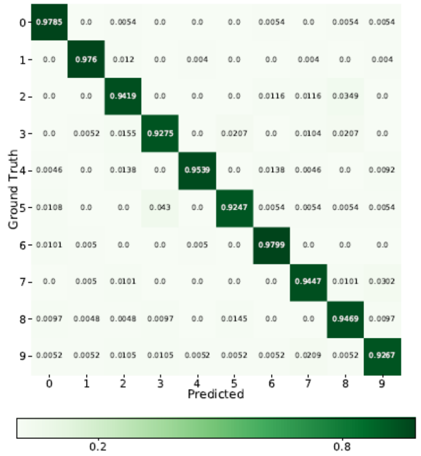
Confusion Matrix



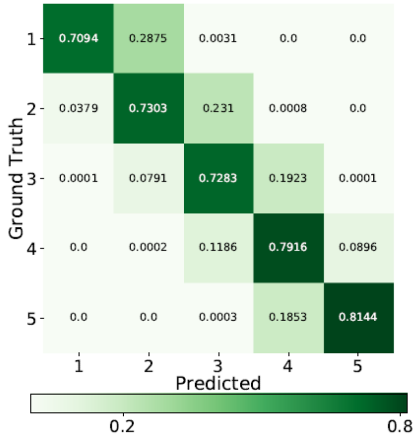
(a) Confusion matrix of PAMAP2



(b) Confusion matrix of TUH

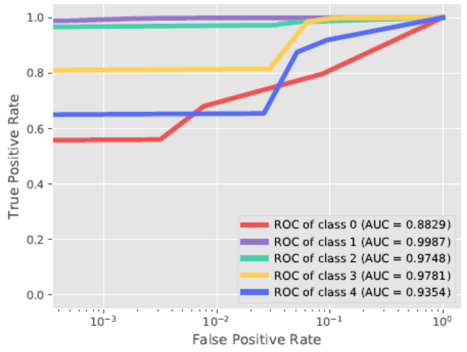


(c) Confusion matrix of MNIST

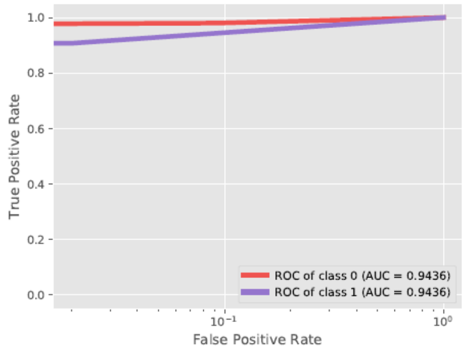


(d) Confusion matrix of Yelp

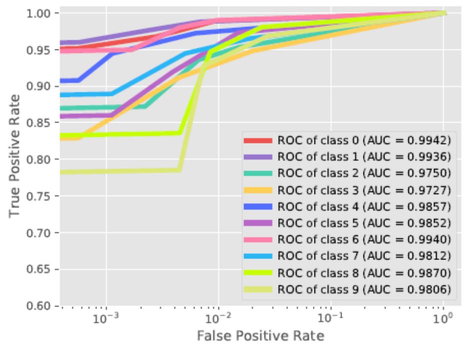
ROC curves



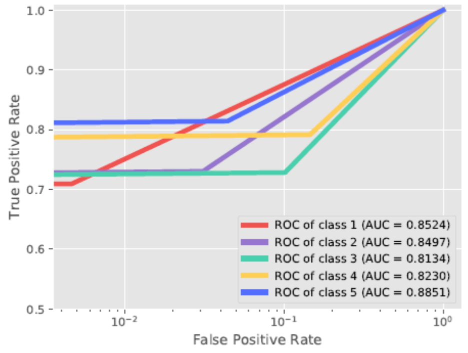
(a) PAMAP2



(b) TUH



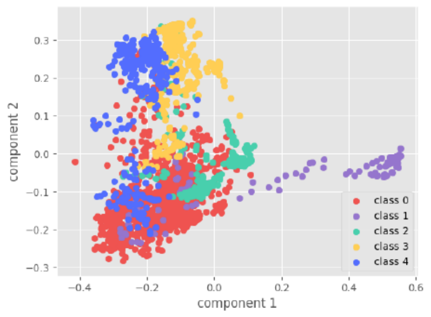
(c) MNIST



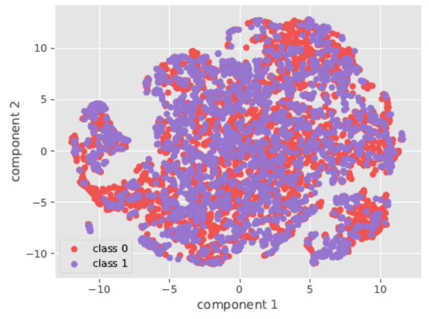
(d) Yelp

서울시립대학교
UNIVERSITY OF SEOUL

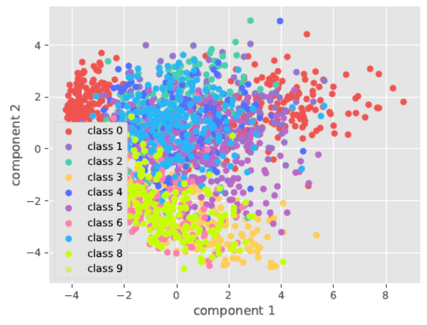
Visualization Comparison



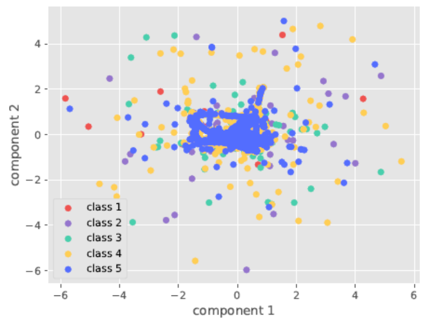
(a) Raw (PAMAP2)



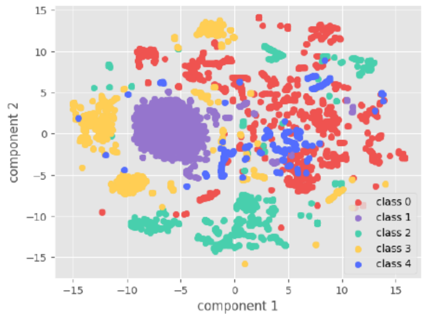
(b) Raw (TUH)



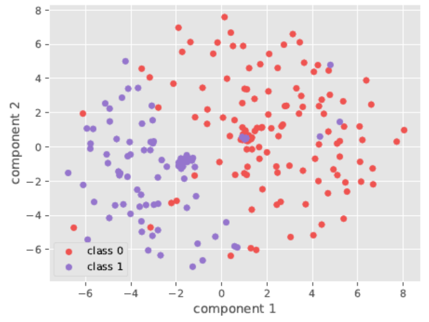
(c) Raw (MNIST)



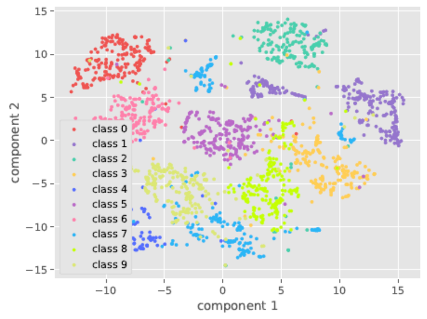
(d) Raw (Yelp)



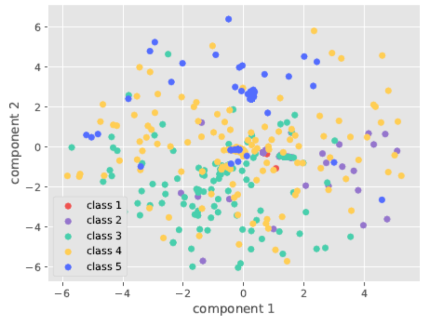
(e) Learned feature (PAMAP2)



(f) Learned feature (TUH)



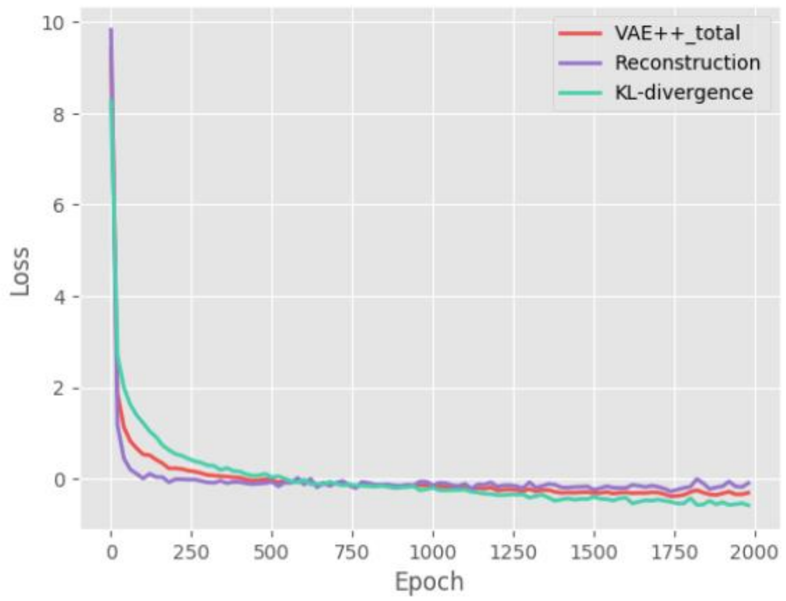
(g) Learned feature (MNIST)



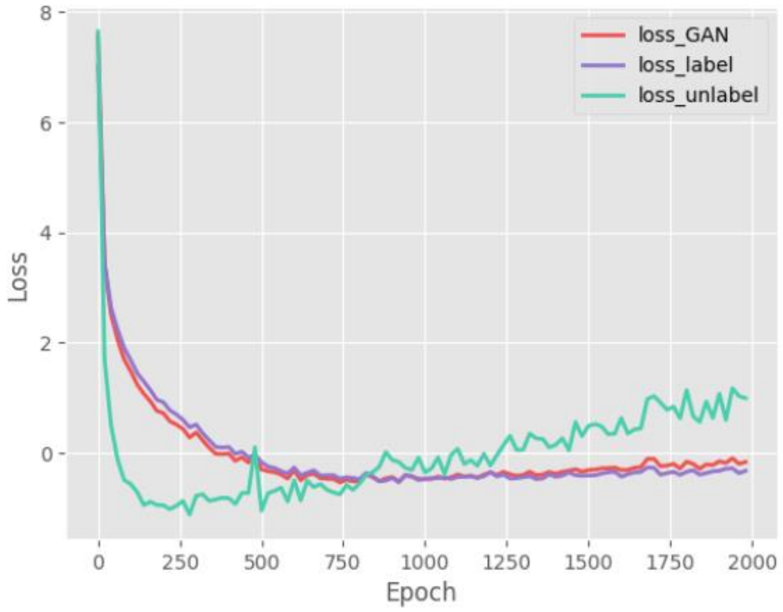
(h) Learned feature (Yelp)

서울시립대학교 UNIVERSITY OF SEOUL

Convergence Curve



(a) VAE++



(b) GAN