Unsupervised Feature Learning for Output Control of Generative Models

Kazuki Toda*, Kyohei Atarashi*, Satoshi Oyama†‡, and Masahito Kurihara‡
*Graduate School of Information Science and Technology, Hokkaido University
†Global Institution for Collaborative Research and Education, Hokkaido University
‡Faculty of Information Science and Technology, Hokkaido University
Kita 14, Nishi 9, Kita-ku, Sapporo, Hokkaido 060-0814, Japan
toda@complex.ist.hokudai.ac.jp, {katarashi, oyama, kurihara}@ist.hokudai.ac.jp

***Abstract*—Deep generative models are being actively studied, particularly variational autoencoders (VAEs) because they can generate high-quality images. The M2 model supports semi-supervised learning from both labeled and unlabeled data, which enables the generated images to be easily controlled by changing the class label values. However, generative models must be learned from only unlabeled data when class labels are not available. A model is presented that incorporates a deep clustering method into the M2 model, which enables clusters to be identified among unlabeled data so that each data point can be assigned to one of the clusters. The generated images in unlabeled datasets can easily be controlled by changing the cluster assignment of each data point.**

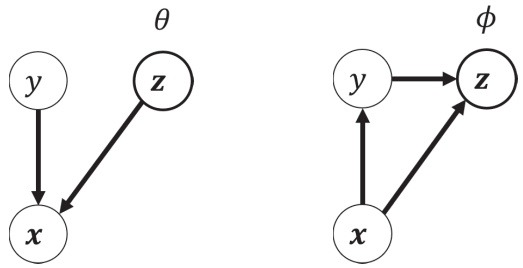
***Index Terms*—unsupervised learning, deep learning, generative model, output control**

I. Introduction

Variational autoencoder (VAE) [9] models generate high-quality images. They are based on the assumption that the input images are generated through latent variables and model the generation process. They have the same the architecture as conventional autoencoders, so they can extract the latent variables of data. We assume that the latent variables have a prior distribution and that new data can be generated by sampling from the prior distribution. However, it is difficult to change the generated images by exploring the latent space because they use continuous priors.

One proposed approach to solving this problem it to use a discrete prior. The Gumbel-softmax [7] method makes the representation of latent features compact and easy to interpret. However, using this method to discretize the latent variable reduces the expression ability of the model. Furthermore, VAEs using the Gumbel-softmax method do not always learn the latent variable that can be used to easily change the generated results.

Another proposed approach is to extend the standard VAE model to semi-supervised learning, enabling the model to learn from both labeled and unlabeled data. The M2 model [8] learns both a generative model and a discriminative model. Using the M2 model, we can change the generated images in a supervised dataset by changing the class label. However, this model cannot learn the classifier in an unsupervised dataset, so we cannot change the generated images by using an unsupervised dataset.



(a) Generative model, p_θ (b) Inference model, q_ϕ

Fig. 1: Generative and inference M2 models.

We propose extending the M2 model to enable control of the generated images for unsupervised learning. The model proposed has the same architecture as the M2 model and incorporates a deep clustering method in the classifier for unsupervised learning. This paper makes two contributions.

- We present a model that can change the generated images in unsupervised learning, and
- we evaluate the performance by comparing it with that of the M2 model.

II. M2 model

VAE models generate data $\boldsymbol{x}$ using latent variables $\boldsymbol{z}$. They can sample from approximate posterior distributions while retaining the parameters in the inference model by using a reparameterization trick. Therefore, it is possible to optimize the inference model and the generative model end-to-end by backpropagation.

The M2 generative model [8] extends the standard VAE model to support semi-supervised learning. The M2 model is illustrated in Fig. 1. It models the generation process by using latent variable $\boldsymbol{z}$ and latent class variable y:

$$p(y) = \mathrm{Cat}(y|\pi), \tag{1}$$
$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|0, I), \tag{2}$$
$$\boldsymbol{\mu} = f_1(\boldsymbol{x}; y, \boldsymbol{z}, \theta), \tag{3}$$
$$\boldsymbol{\sigma} = f_2(\boldsymbol{x}; y, \boldsymbol{z}, \theta), \tag{4}$$
$$p_\theta(\boldsymbol{x}|y, \boldsymbol{z}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2), \tag{5}$$

where Cat is a categorical distribution, $\mathcal{N}$ is a Gaussian distribution, and f_1 and f_2 are mapping functions like neural networks. It can use two kinds of data: labeled data and unlabeled data. The objective function consists of a labeled bound, an unlabeled bound, and label loss. The variational bound for labeled data is defined as

$$\begin{aligned}\log p_\theta(\boldsymbol{x}, y) \geq \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x},y)}[&\log p_\theta(\boldsymbol{x}|y,\boldsymbol{z}) + \log p_\theta(y) \\ &+ \log p(\boldsymbol{z}) - \log q_\phi(\boldsymbol{z}|\boldsymbol{x},y)] \quad (6)\\ &= -\mathcal{L}(\boldsymbol{x}, y). \quad (7)\end{aligned}$$

The variational bound for unlabeled data is as

$$\begin{aligned}\log p_\theta(\boldsymbol{x}) \geq \mathbb{E}_{q_\phi(y,\boldsymbol{z}|\boldsymbol{x})}[&\log p_\theta(\boldsymbol{x}|y,\boldsymbol{z}) + \log p_\theta(y) \\ &+ \log p(\boldsymbol{z}) - \log q_\phi(y,\boldsymbol{z}|\boldsymbol{x})] \quad (8)\\ &= \sum_y q_\phi(y|\boldsymbol{x})(-\mathcal{L}(\boldsymbol{x}, y)) + h(q_\phi(y|\boldsymbol{x})) \quad (9)\\ &= -\mathcal{U}(\boldsymbol{x}), \quad (10)\end{aligned}$$

where $h(p(y)) \equiv -\sum_{y'} p(y') \log p(y')$ is the entropy function. Therefore, the bound on the negative log-likelihood for all data is

$$\mathcal{J} = \sum_{(\boldsymbol{x},y)\sim\tilde{p}_l} \mathcal{L}(\boldsymbol{x}, y) + \sum_{\boldsymbol{x}\sim\tilde{p}_u} \mathcal{U}(\boldsymbol{x}), \quad (11)$$

where $\tilde{p}$ is an empirical distribution.

In objective function (11), the predicted label distribution $q_\phi(y|\boldsymbol{x})$ appears only in unlabeled terms. Thus, it is impossible to learn using label information for learning the classifier. This problem is overcome by using a newly proposed objective function in which classification loss is added to (11).

The extended objective function is

$$\mathcal{J}^\alpha = \mathcal{J} + \alpha \cdot \mathbb{E}_{\tilde{p}_l(\boldsymbol{x},y)}[-\log q_\phi(y|\boldsymbol{x})], \quad (12)$$

where α is a hyperparameter for controlling the balance between generative and discriminative learning.

III. IMSAT

The Information Maximizing Self-Augmented Training (IMSAT) [5] method is used for clustering or hash learning using a deep neural network. In this paper, we focus on IMSAT for clustering, which combines regularized information maximization (RIM) [4] for clustering and self-augmented training (SAT) [13].

A. Regularized Information Maximization

The RIM [4] method improves discrimination performance by maximizing the mutual information between the inputs and cluster assignments. It also controls the complexity of the classifier by regularization.

Let K be the number of clusters, $\mathcal{X}$ be the input domain, and $\mathcal{Y}$ be the discrete representation domain. In addition, let $X \in \mathcal{X}$ be a random variable for input data and $Y \in \mathcal{Y} \equiv \{0, ..., K-1\}$ be one for cluster assignment. We can now minimize the objective function for clustering to

$$\mathcal{R}(\theta) - \lambda I(X; Y), \quad (13)$$

where $\mathcal{R}(\theta)$ is the regularization term and $I(X;Y)$ is the mutual information term between input X and cluster variable Y. Both depend on parameter θ. Hyperparameter λ controls the balance between these terms.

B. Self-Augmented Training

SAT [13] uses data augmentation to make the representation locally invariant. This brings the augmented data points close to the original ones. Let the augmented function be $T : X \to \mathcal{X}$. Then, the regularization at a data point $\boldsymbol{x}_n$ is

$$\begin{aligned}&\mathcal{R}_{\mathrm{SAT}}(\theta; \boldsymbol{x}_n, T(\boldsymbol{x}_n)) = \\ &\quad -\sum_{m=1}^{M}\sum_{y_m=0}^{V_m} p_{\hat{\theta}}(y_m|\boldsymbol{x}_n) \log p_\theta(y_m|T(\boldsymbol{x}_n)), \quad (14)\end{aligned}$$

where $\hat{\theta}$ is the current network parameter, and $p_{\hat{\theta}}$ is the prediction of original data point $\boldsymbol{x}$. Finally, the objective function is the average of equation (14) at all training data points:

$$\mathcal{R}(\theta, T) = \frac{1}{N}\sum_{n=1}^{N} \mathcal{R}_{\mathrm{SAT}}(\theta; \boldsymbol{x}_n, T(\boldsymbol{x}_n)). \quad (15)$$

There is then a general augmented function T that adds a local perturbation. The augmented function is expressed as

$$T(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{r}, \quad (16)$$

where $\boldsymbol{r}$ is slight perturbation that does not change the average of the data points. One of the methods for determining this perturbation r is SAT, which selects the most hostile perturbation in accordance with

$$\boldsymbol{r} = \arg\max_{\boldsymbol{r}'}\{\mathcal{R}_{\mathrm{SAT}}(\hat{\theta}; \boldsymbol{x}, \boldsymbol{x} + \boldsymbol{r}'); \|\boldsymbol{r}'\|_2 \leq \epsilon\}. \quad (17)$$

C. IMSAT for clustering

IMSAT for clustering combines RIM with SAT. The objective function is

$$\begin{aligned}&\mathcal{R}_{\mathrm{SAT}} - \lambda[I(X;Y)] = \\ &\mathcal{R}_{\mathrm{SAT}} - \lambda[H(Y) - H(Y \mid X)], \quad (18)\end{aligned}$$

where $H(\cdot)$ is the entropy and $H(\cdot \mid \cdot)$ is the conditional entropy. These terms are calculated using

$$H(Y) \equiv h(p_\theta(y)) = h\left(\frac{1}{N}\sum_{i=1}^{N} p_\theta(y|\boldsymbol{x}_i)\right), \quad (19)$$

$$H(Y \mid X) \equiv \frac{1}{N}\sum_{i=1}^{N} h(p_\theta(y \mid \boldsymbol{x}_i)). \quad (20)$$

The second term in (18) makes the distribution at whole data points uniform, and the third term makes the predicted distribution at each data point sharp.

IV. Related work

It is important to use discrete prior distributions because doing so improves the interpretability and compactness of latent variables. However, it is impossible to use discrete prior distributions with end-to-end learning since the reparameterization trick [9] supports only continuous distributions. To solve this problem, Eric Jang et al. proposed using approximate sampling from a categorical distribution. Their Gumbel-softmax method [7] not only solves this problem but also makes learning faster than marginalizing labels in the M2 model.

The vector-quantized (VQ)-VAE [15] and the self-organizing-map (SOM)-VAE [3] extensions of VAE discretize the latent variables. VQ-VAE applies vector quantization to the latent variables and embeds them in embedding space, thereby reducing the number of inactive units. SOM-VAE, a generalized VQ-VAE, applies self-organizing maps [10] to the latent variables. In addition, it uses a Markov model, making it easier to learn series data. It thus improves the interpretability of the latent variables and can be applied to complex time-series data such as medical data.

The Cluster-aware Generative Model [12] has two layers of latent variables and uses natural clustering to extract the data structure. Its use of natural clustering improves generative performance.

Thomas et al. proposed an objective function [14] for learning feature expressions that can be used to control data changes. It consists of a reconstruction error term in the autoencoder and a *selectivity* term. The selectivity term makes the feature expressions learned by the autoencoder controllable and independent of the elements.

V. Proposed model

Our proposed model for unsupervised feature learning introduces IMSAT clustering into the M2 model, making it possible to control the generated images by changing the cluster assignments even when learning from unlabeled datasets. This model has the same architecture as the M2 model except for introducing IMSAT clustering into the classifier. The objective function for unsupervised learning is

$$\begin{aligned}\mathcal{K} = &- \sum_{x \sim \tilde{p_u}} \mathcal{U}(\boldsymbol{x}) \\ &+ \gamma(\mathcal{R}_{\mathrm{SAT}} - \lambda I(X;Y)),\end{aligned} \tag{21}$$

where γ is a hyperparameter that controls the power of the IMSAT terms.

The proposed model can also be used for semi-supervised learning by using both labeled and unlabeled data. The objective function for semi-supervised learning is

$$\mathcal{K}^{\alpha} = \mathcal{K} - \sum_{\boldsymbol{x},y \sim \tilde{p}_l} \mathcal{L}(\boldsymbol{x}, y) + \alpha \cdot \mathbb{E}_{\boldsymbol{x},y \sim \tilde{p}_l}[\log q_\phi(y \mid \boldsymbol{x})]. \tag{22}$$

The generative and clustering tasks are done by minimizing either of these objective functions.

VI. Evaluation

We evaluated our proposed model in comparison with the M2 model using three benchmark datasets and two quantitative evaluation measures, the ELBO (evidence lower bound) score and labeling/clustering accuracy. The ELBO score was calculated on the basis of the log-likelihood scores given by the importance weighted autoencoder (IWAE). The labeling/clustering accuracy was used to evaluate the similarity of the predicted labels or clusters to the true labels and was calculated using

$$\max_{m} \frac{\sum_{n=1}^{N} \mathbf{1}\{l_n = m(c_n)\}}{N}, \tag{23}$$

where N is the number of classes, l_n is true labels, and c_n is assigned clusters. As a qualitative evaluation, we evaluated reconstructed images of training and test data and evaluated generated images when the cluster assignments and latent variables were changed.

A. Experiments

We conducted four qualitative evaluation experiments using the MNIST (Mixed National Institute of Standards and Technology) [11], Kuzushiji-MNIST (KMNIST) [1], and Fashion-MNIST (FMNIST) [16] datasets. The KMNIST dataset consists of ten Japanese hiragana characters, and the FMNIST one consists of ten fashion items. Both have the same form as the MNIST one but are more complicated. In each experiment, we used 60,000 images as training data and 10,000 images as test data.

Since the proposed model has the same architecture as the M2 model, we used the same settings for both. We used the maximum likelihood probability for the encoders, decoders, and classifiers. We set the encoder network dimensionality to 784-500-50 and the decoder network one to 60-500-784. We set the classifier network one to 784-500-10 for the MNIST dataset and to 784-500-500-2000-10 for the other datasets. We used a softplus function for the hidden activations in the encoders and decoders. In addition, we used softplus [2] as the activation function in the classifiers and applied batch normalization [6] to each layer.

We set the mini-batch size to 200. For semi-supervised learning, the mini-batch consisted of 100 labeled data points and 100 unlabeled data points. We trained the M2 model and our model for 500 epochs.

We use the Gumbel-softmax method to accelerate learning. It enables approximation of sampling from a discrete distribution. We fixed temperature parameter τ to 0.1 for learning with the MNIST dataset, initialized it for this method to 5, and annealed it 0.99 times per epoch.

We set parameter μ for controlling the power of marginal entropy $H(Y)$ to four and λ for controlling the balance between $\mathcal{R}_{\mathrm{SAT}}$ and $I(X;Y)$ to 0.1 in all experiments. We used VAT as the augmented function and set parameter α so that the range of perturbation was 0.25. Finally, we set γ for adjusting the balance between the IMSAT term and ELBO to 1000.

TABLE I: ELBO scores

Model version	MNIST	KMNIST	FMNIST
M2 (supervised)	-93.476	-192.224	-235.269
M2 (unsup.)	-95.279	-191.041	-234.711
M2 (semi-sup.)	-116.086	-360.480	-267.816
Our model (unsup.)	-93.620	-190.575	-234.449
Our model (semi-sup)	-141.819	-360.343	-244.107

TABLE II: Labeling/Clustering accuracy for training data.

Model version	MNIST	KMNIST	FMNIST
M2 (unsup.)	0.484	0.558	0.507
M2 (semi-sup.)	0.943	0.620	0.687
Our model (unsup.)	0.821	0.679	0.645
Our model (semi-sup.)	0.979	0.718	0.749

TABLE III: Labeling/Clustering accuracy for test data

Model	MNIST	KMNIST	FMNIST
M2 (unsup.)	0.484	0.372	0.500
M2 (semi-sup.)	0.942	0.463	0.681
Our model (unsup.)	0.825	0.503	0.636
Our model (semi-sup.)	0.981	0.553	0.740

B. Results

1) ELBO scores: We calculated the ELBO scores for the supervised version of the M2 model by using test data with true labels and for the unsupervised and semi-supervised version of the M2 model and our model by using test data with only the predicted labels (cluster assignment). The unsupervised and semi-supervised versions of our model have the same structures as those of the M2 model. Table I summarizes the ELBO scores. The higher the score, the better the performance because the ELBO score is a lower bound of the log-likelihood. Both versions of the proposed model performed better than those of the original one except for the semi-supervised version when trained with MNIST. This is because the MNIST dataset is simple, so data augmentation by IMSAT clustering did not benefit the training of the generative model. In fact, it negatively affected it.

The scores of the semi-supervised learning versions of the models were lower than those of the unsupervised ones. This is because the unsupervised versions learned only a generation task while the semi-supervised ones learned both classification and generation tasks. In addition, the ELBO score reflects the performance of only the generation task. Thus, the semi-supervised versions did not always achieve good optimization for the generation task.

2) Labeling/Clustering accuracy: We calculated the labeling/clustering values for both versions of the M2 model and our model. Performance is high when the value is close to 1. As shown in Tables II and III, IMSAT clustering improved accuracy with and without labeling/clustering. In addition, the semi-supervised learning version of our model had higher accuracy than that of the M2 model trained using the MNIST dataset. Therefore, our model is effective when classification by shape is important.

The accuracy for the test data in the KMNIST dataset was worse than the accuracy for the test data in the other datasets. This is because the data in the KMNIST dataset is more difficult to classify than those in the other datasets as there is a variety of images in the same class (character), and there is sometimes a small difference between images in different classes.

3) Qualitative evaluation of generated images: For qualitative evaluation of the generated images, we fixed a latent variable in each version of the two models and changed only the label/cluster variables to enable evaluation of whether each version can learn discrete variables for effectively changing generated images. The training and generated MNIST images are shown in Fig. 2. The results for the unsupervised learning version of the M2 model show that it failed to generate correct images for several classes.

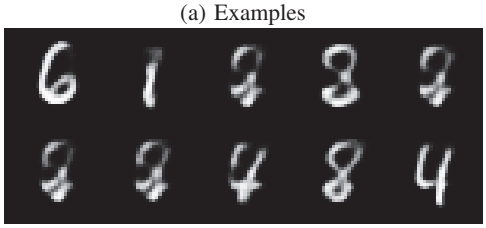
The training and generated KMNIST images are shown in Fig. 3 These results show that the unsupervised version of our model improved the quality of the generated results while the unsupervised version of the M2 model did not successfully generate images for some classes. Therefore, classification by shape needs to be successfully performed in order to change the generated images. In addition, the images generated by the semi-supervised version of the M2 model and the semi-supervised version of our model are ambiguous. It is difficult to learn the classifier from a small number of images for each label because the images in the KMNIST dataset for each label have greatly different shapes. Therefore, given the generated images and the results in Table I, we assume that the semi-supervised learning versions of both models overfit the shape of the labeled data, resulting in generated images with poor quality.

These results show that the unsupervised versions of the M2 model and our model could not recognize some classes and thus failed to generate images for those classes. This is because the FMNIST dataset contains many similar images belonging to different classes, which makes it difficult to classify them using only visual information. However, it is easier to change the generated images with the unsupervised version of our model because the IMSAT clustering improves classification performance. In contrast, the semi-supervised versions of the M2 model and our model solve this problem and successfully generate images for all classes. Therefore, label information is effective for guiding the learning of the model so that it can distinguish different classes. In addition, the semi-supervised version of our model generated smoother images than the semi-supervised version of the M2 model. This is because the IMSAT clustering augmented the data for each class, enabling the model to learn the generative process of each class more accurately.

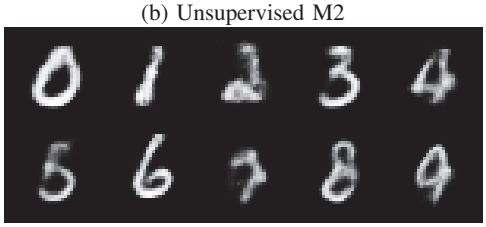
VII. Summary and future work

Our proposed generative model combines the M2 model with IMSAT clustering, so the generated images can be easily controlled by changing the label (cluster assignment) with both unsupervised and semi-supervised learning. Comparison of evaluation results between our model and the M2 model showed that the proposed model can learn labels/clusters

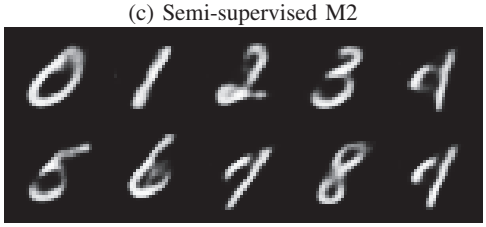
(a) Examples



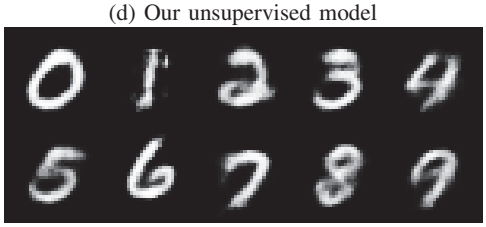
(b) Unsupervised M2



(c) Semi-supervised M2



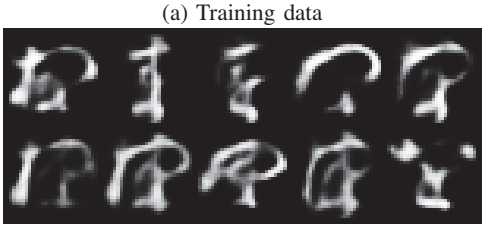
(d) Our unsupervised model



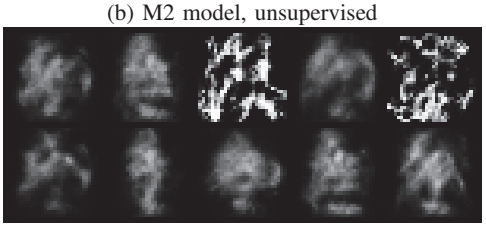
(e) Our semi-supervised model

Fig. 2: MNIST images of each label in (a) training data, and generated by (b) unsupervised M2 model, (c) semi-supervised M2 model, (d) our unsupervised model, and (e) our semi-supervised model.

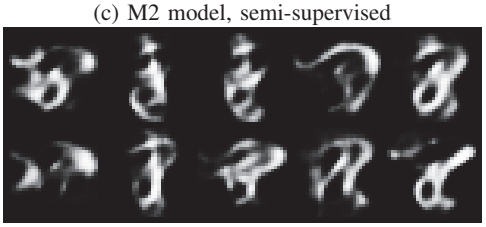
(a) Training data



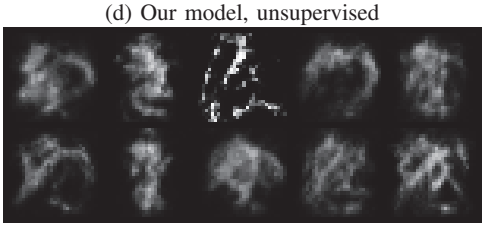
(b) M2 model, unsupervised



(c) M2 model, semi-supervised



(d) Our model, unsupervised



(e) Our model, semi-supervised

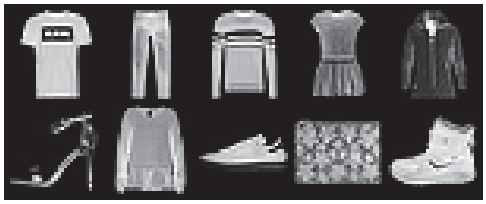
Fig. 3: KMNIST images for each label for (a) training data and generated by (b) unsupervised version of M2 model, (c) semi-supervised version of M2 model, (d) unsupervised version of our model, and (e) semi-supervised version of our model.

for effectively changing the generated images. The results also showed that IMSAT clustering improves classification accuracy, especially for datasets where classification by shape is important, like the MNIST and KMNIST datasets. These results indicate that the proposed model is particularly effective when classification by shape is important. The results for the KMNIST dataset show that generative models with semi-supervised learning do not always generate better images than ones with unsupervised learning.

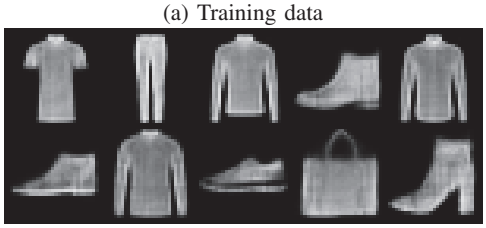
Future work includes changing from IMSAT for clustering to IMSAT for hash learning. While a cluster variable represents a one-hot vector, a hash variable represents a binarized vector. Therefore, the M2 model with IMSAT for hash learning can learn more expressive discrete representations, enabling generated images to be changed more flexibly.

ACKNOWLEDGMENTS

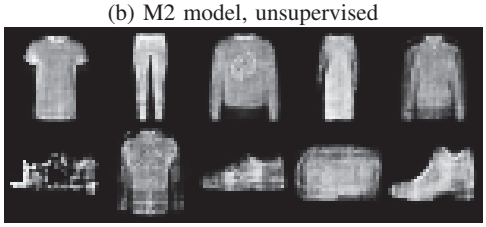
This work was partially supported by JSPS KAKENHI Grant JP18H03337, by the Telecommunications Advancement Foundation, and by the Global Station for Big Data and Cybersecurity, a project of the Global Institution for Collaborative Research and Education at Hokkaido University.



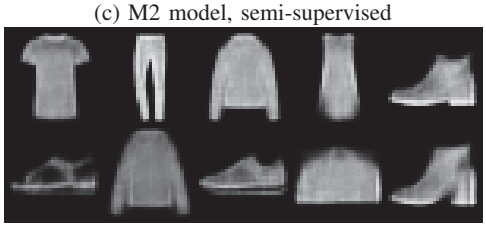
(a) Training data



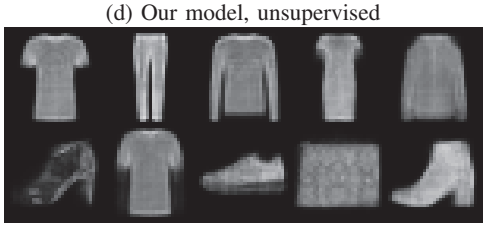
(b) M2 model, unsupervised



(c) M2 model, semi-supervised



(d) Our model, unsupervised



(e) Our model, semi-supervised

Fig. 4: FMNIST images for each label for (a) training data and generated by (b) unsupervised version of M2 model, (c) semi-supervised version of M2 model, (d) unsupervised version of our model, and (e) semi-supervised version of our model.

REFERENCES

[1] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

[2] Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pages 472–478. MIT Press, 2000.

[3] Vincent Fortuin, Matthias Hüser, Francesco Locatello, Heiko Strathmann, and Gunnar Rätsch. SOM-VAE: interpretable discrete representation learning on time series. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[4] Ryan Gomes, Andreas Krause, and Pietro Perona. Discriminative clustering by regularized information maximization. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 775–783, 2010.

[5] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 1558–1567, 2017.

[6] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 448–456, 2015.

[7] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[8] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[9] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[10] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[11] Yann LeCun, Léon. Bottou, Yoshua Bengio, and Patrick Haffener. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[12] Lars Maaløe, Marco Fraccaro, and Ole Winther. Semi-supervised generation with cluster-aware generative models. *arXiv preprint arXiv:1704.00637*, 2017.

[13] Takeru Miyato, Shinichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. Distributional smoothing with virtual adversarial training. *arXiv preprint arXiv:1507.00677*, 2015.

[14] Valentin Thomas, Jules Pondard, Emmanuel Bengio, Marc Sarfati, Philippe Beaudoin, Marie-Jean Meurs, Joelle Pineau, Doina Precup, and Yoshua Bengio. Independently controllable factors. *arXiv preprint arXiv:1708.01289*, 2017.

[15] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6306–6315, 2017.

[16] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.