Deep Clustering by Gaussian Mixture Variational Autoencoders with Graph Embedding - Supplementary

Linxiao Yang*[1,2], Ngai-Man Cheung‡[1], Jiaying Li[1], and Jun Fang[2]

[1]Singapore University of Technology and Design (SUTD)
[2]University of Electronic Science and Technology of China

‡Corresponding author: ngaiman_cheung@sutd.edu.sg

1. Proof of equation (13)

We provide detail to show how to arrive at (13)

$$\begin{aligned}
G(\phi,\theta,\boldsymbol{x}_i,\boldsymbol{x}_j) =& \frac{1}{2}\int q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_i)\ln\frac{q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_i)}{p_\theta(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_i)}d\boldsymbol{z}d\boldsymbol{c} + \frac{1}{2}\int q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)\ln\frac{q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)}{p_\theta(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_i)}d\boldsymbol{z}d\boldsymbol{c} \\
=& \frac{1}{2}\int q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_i)\ln\frac{q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_i)}{M}d\boldsymbol{z}d\boldsymbol{c} + \frac{1}{2}\int q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)\ln\frac{q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)}{M}d\boldsymbol{z}d\boldsymbol{c} \\
&+ \int\frac{1}{2}(q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_i)+q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j))\ln\frac{M}{p_\theta(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_i)}d\boldsymbol{z}d\boldsymbol{c} \\
=& \mathrm{JS}(q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_i)||q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)) + \mathrm{KL}(M||p_\theta(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_i)) \\
\geq& \mathrm{JS}(q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_i)||q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j))
\end{aligned} \tag{1}$$

2. Proof of equation (15)

In this part, we provide proof to show that that we can replace $q(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_i)$ with $q(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)$ in (14) i.e. the decomposition of log-likelihood $\ln p_\theta(\boldsymbol{x}_i)$

$$\begin{aligned}
&\mathrm{KL}(q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)||p_\theta(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_i)) + E_{q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)}\left[\ln\frac{p_\theta(\boldsymbol{x}_i,\boldsymbol{z},\boldsymbol{c})}{q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)}\right] \\
=& -\int q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)\ln\frac{p_\theta(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_i)}{q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)}d\boldsymbol{z}d\boldsymbol{c} + \int q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)\ln\frac{p_\theta(\boldsymbol{x}_i,\boldsymbol{z},\boldsymbol{c})}{q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)}d\boldsymbol{z}d\boldsymbol{c} \\
=& \int q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)\left(\ln\frac{p_\theta(\boldsymbol{x}_i,\boldsymbol{z},\boldsymbol{c})}{q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)} - \ln\frac{p_\theta(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_i)}{q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)}\right)d\boldsymbol{z}d\boldsymbol{c} \\
=& \int q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)\left(\ln\frac{p_\theta(\boldsymbol{x}_i,\boldsymbol{z},\boldsymbol{c})q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)}{q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)p_\theta(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_i)}\right)d\boldsymbol{z}d\boldsymbol{c} \\
=& \int q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)\ln p_\theta(\boldsymbol{x}_i)d\boldsymbol{z}d\boldsymbol{c} \\
=& \ln p_\theta(\boldsymbol{x}_i)\int q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j)d\boldsymbol{z}d\boldsymbol{c} \\
=& \ln p_\theta(\boldsymbol{x}_i)
\end{aligned} \tag{2}$$

*Work done during an internship at SUTD

3. Derivation of equation (19)

We provide detail to show how to arrive at (19) from (11)

$$\begin{aligned}
&\sum_{i=1}^{N}\left(\ln p_\theta(\boldsymbol{x}_i)-\sum_{j=1}^{N} w_{ij} JS(q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_i), q_\phi(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x}_j))\right)\\
\overset{(a)}{\geq}&\sum_{i=1}^{N}\left(\ln p_\theta(\boldsymbol{x}_i)-\sum_{j=1}^{N} w_{ij} G(\phi,\theta,\boldsymbol{x}_i,\boldsymbol{x}_j)\right)\\
\overset{(b)}{=}&\sum_{i=1}^{N}\left(\ln p_\theta(\boldsymbol{x}_i)-\sum_{j=1}^{N} w_{ij} \ln p_\theta(\boldsymbol{x}_i)\right)+\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}(L(\boldsymbol{\theta},\boldsymbol{\phi};\boldsymbol{x}_i)+L(\boldsymbol{\theta},\boldsymbol{\phi};\boldsymbol{x}_i,\boldsymbol{x}_j))\\
\overset{(c)}{=}&\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}(L(\boldsymbol{\theta},\boldsymbol{\phi};\boldsymbol{x}_i)+L(\boldsymbol{\theta},\boldsymbol{\phi};\boldsymbol{x}_i,\boldsymbol{x}_j))
\end{aligned}\tag{3}$$

where inequation (a) comes from (13), equation (b) comes from (16), and equation (c) comes from the fact that $\sum_j w_{ij}=1$.

4. Derivation of equation (21)

We provide details to show how $L(\theta,\phi,\boldsymbol{x}_i)$ can be estimated using (21). We note that the equation (26) can be derived similarly. The equation (21) can be written as

$$\begin{aligned}
&L(\boldsymbol{\theta},\boldsymbol{\phi};\boldsymbol{x}_i)\\
=&E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)q_\phi(\boldsymbol{c}|\boldsymbol{z})}\left[\ln\frac{p_\theta(\boldsymbol{x}_i|\boldsymbol{z})p(\boldsymbol{z}|\boldsymbol{c})p(\boldsymbol{c})}{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)q_\phi(\boldsymbol{c}|\boldsymbol{z})}\right]\\
=&E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)q_\phi(\boldsymbol{c}|\boldsymbol{z})}\left[\ln p_\theta(\boldsymbol{x}_i|\boldsymbol{z})+\ln p(\boldsymbol{z}|\boldsymbol{c})+\ln p(\boldsymbol{c})-\ln q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)-\ln q_\phi(\boldsymbol{c}|\boldsymbol{z})\right]\\
=&E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)}[\ln p_\theta(\boldsymbol{x}_i|\boldsymbol{z})]+E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)q_\phi(\boldsymbol{c}|\boldsymbol{z})}[\ln p(\boldsymbol{z}|\boldsymbol{c})]+E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)q_\phi(\boldsymbol{c}|\boldsymbol{z})}\left[\ln\frac{p(\boldsymbol{c})}{q_\phi(\boldsymbol{c}|\boldsymbol{z})}\right]-E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)}[\ln q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)]
\end{aligned}\tag{4}$$

In the following, we evaluate the terms in (4).

4.1. Evaluate $E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)}[\ln p_\theta(\boldsymbol{x}_i|\boldsymbol{z})]$

The term $E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)}[\ln p_\theta(\boldsymbol{x}_i|\boldsymbol{z})]$ can be estimated using the reparameterization trick, i.e.,

$$E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)}[\ln p_\theta(\boldsymbol{x}_i|\boldsymbol{z})]\approx\frac{1}{L}\sum_{l=1}^{L}\log p(\boldsymbol{x}|\boldsymbol{z}_i^{(l)})\tag{5}$$

where $\boldsymbol{z}_i^{(l)}\sim\mathcal{N}(\boldsymbol{\mu},\text{diag}(\boldsymbol{\sigma}^2))$, and $[\tilde{\boldsymbol{\mu}},\log\tilde{\boldsymbol{\sigma}}^2]=f(\boldsymbol{x}_i,\phi)$. L is the number of samples that used in the SGVB estimator. In our method, we set $L=1$ and omit the superscript of $\boldsymbol{z}_i$. Then the term $E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)}[\ln p_\theta(\boldsymbol{x}_i|\boldsymbol{z})]$ can be estimated as

$$E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)}[\ln p_\theta(\boldsymbol{x}_i|\boldsymbol{z})]\approx\begin{cases}\sum_{d=1}^{D} x_d^i\log\boldsymbol{\mu}_{x_i}|_d+(1-x_d^i)\log(1-\boldsymbol{\mu}_{x_i}|_d) & \text{if } \boldsymbol{x}_i \text{ is binary}\\ \frac{1}{2\lambda}\|\boldsymbol{x}_i-\boldsymbol{\mu}_{x_i}\|_2^2 & \text{if } \boldsymbol{x}_i \text{ is real-valued}\end{cases}\tag{6}$$

where $\boldsymbol{\mu}_{x_i}=g(\boldsymbol{z}_i;\theta)$.

4.2. Evaluate $E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)q_\phi(\boldsymbol{c}|\boldsymbol{z})}[\ln p(\boldsymbol{z}|\boldsymbol{c})]$

We note that the term $E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)q_\phi(\boldsymbol{c}|\boldsymbol{z})}[\ln p(\boldsymbol{z}|\boldsymbol{c})]$ can be estimated using the reparameterization trick. But in our paper, we use an approximation of this term, i.e., we approximate $q(\boldsymbol{c}|\boldsymbol{z})$ using $q(\boldsymbol{c}|\boldsymbol{z})$. Then

$$
\begin{aligned}
E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)q_\phi(\boldsymbol{c}|\boldsymbol{z})}[\ln p(\boldsymbol{z}|\boldsymbol{c})] \approx & E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)q_\phi(\boldsymbol{c}|\boldsymbol{z}_i)}[\ln p(\boldsymbol{z}|\boldsymbol{c})] && (7)\\
= & E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)q_\phi(\boldsymbol{c}|\boldsymbol{z}_i)}\left[\ln\left(\prod_{k=1}^{K}\mathcal{N}(\boldsymbol{\mu}_k,\mathrm{diag}(\boldsymbol{\sigma}_k^2))^{c_k}\right)\right] && (8)\\
= & E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)q_\phi(\boldsymbol{c}|\boldsymbol{z}_i)}\left[\sum_{k=1}^{K}c_k\ln\mathcal{N}(\boldsymbol{\mu}_k,\mathrm{diag}(\boldsymbol{\sigma}_k^2))\right] && (9)\\
= & \sum_{k=1}^{K}\gamma_{ik}E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)}\left[\ln\mathcal{N}(\boldsymbol{\mu}_k,\mathrm{diag}(\boldsymbol{\sigma}_k^2))\right] && (10)\\
= & \sum_{k=1}^{K}\gamma_{ik}E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)}\left[\sum_{m=1}^{M}(-\frac{1}{2}\ln\boldsymbol{\sigma}_k^2|_m-\frac{(\boldsymbol{z}|_m-\boldsymbol{\mu}_k|_m)^2}{2\boldsymbol{\sigma}_k^2|_m})\right] && (11)
\end{aligned}
$$

As $q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)$ is Gaussian distribution with mean $\tilde{\boldsymbol{\mu}}_i$ and corvariance matrix $\mathrm{diag}(\tilde{\boldsymbol{\sigma}}_i^2)$, where $[\tilde{\boldsymbol{\mu}}_i,\log(\tilde{\boldsymbol{\sigma}}_i^2)]=f_1(\boldsymbol{x}_i;\phi_1)$, we have

$$
\begin{aligned}
E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)}[\boldsymbol{z}|_m] &= \tilde{\boldsymbol{\mu}}_i|_m^2 && (12)\\
E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)}[\boldsymbol{z}|_m^2] &= \tilde{\boldsymbol{\mu}}_i|_m^2+\tilde{\boldsymbol{\sigma}}_i^2|_m && (13)
\end{aligned}
$$

Substituting the above equations into (11), we arrive at

$$
E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)q_\phi(\boldsymbol{c}|\boldsymbol{z})}[\ln p(\boldsymbol{z}|\boldsymbol{c})] \approx -\frac{1}{2}\sum_{k=1}^{K}\gamma_{ik}\sum_{m=1}^{M}(\ln\boldsymbol{\sigma}_k^2|_m+\frac{\tilde{\boldsymbol{\sigma}}_i^2|_m}{\boldsymbol{\sigma}_k^2|_m}+\frac{(\tilde{\boldsymbol{\mu}}_i|_m-\boldsymbol{\mu}_k|_m)^2}{\boldsymbol{\sigma}_k^2|_m}) \quad (14)
$$

4.3. Evaluate $E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)q_\phi(\boldsymbol{c}|\boldsymbol{z})}\left[\ln\frac{p(\boldsymbol{c})}{q_\phi(\boldsymbol{c}|\boldsymbol{z})}\right]$

The term $E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)q_\phi(\boldsymbol{c}|\boldsymbol{z})}\left[\ln\frac{p(\boldsymbol{c})}{q_\phi(\boldsymbol{c}|\boldsymbol{z})}\right]$ can be estimated by

$$
\begin{aligned}
E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)q_\phi(\boldsymbol{c}|\boldsymbol{z})}\left[\ln\frac{p(\boldsymbol{c})}{q_\phi(\boldsymbol{c}|\boldsymbol{z})}\right] \approx & E_{q_\phi(\boldsymbol{c}|\boldsymbol{z}_i)}\left[\ln\frac{p(\boldsymbol{c})}{q_\phi(\boldsymbol{c}|\boldsymbol{z}_i)}\right] && (15)\\
= & E_{q_\phi(\boldsymbol{c}|\boldsymbol{z}_i)}\left[\sum_{k=1}^{K}c_k\ln\frac{\pi_{ik}}{\gamma_{ik}}\right] && (16)\\
= & \sum_{k=1}^{K}\gamma_{ik}\ln\frac{\pi_{ik}}{\gamma_{ik}} && (17)
\end{aligned}
$$

where (15) comes from the reparameterization trick, γ_{ik} denotes the kth entry of $q_\phi(\boldsymbol{c}|\boldsymbol{z}_i)$, and (17) comes from $E_{q_\phi(\boldsymbol{c}|\boldsymbol{z}_i)}[c_k]=\gamma_{ik}$.

4.4. Evaluate $E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)}[\ln q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)]$

We evaluate $E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)}[\ln q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)]$. As mentioned above, $q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)$ is a Gaussian distribution. Then the term $E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)}[\ln q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)]$ can be rewritten as

$$\begin{aligned} E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)}[\ln q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)] =& E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)}\left[\sum_{m=1}^{M}\left(-\frac{1}{2}\log\tilde{\boldsymbol{\sigma}}_i^2|_m - \frac{(\boldsymbol{z}|_m - \tilde{\boldsymbol{\mu}}_i|_m)^2}{2\tilde{\boldsymbol{\sigma}}_i^2|_m}\right)\right] \\ =& \sum_{m=1}^{M}\left(-\frac{1}{2}\log\tilde{\boldsymbol{\sigma}}_i^2|_m - \frac{E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)}\left[\boldsymbol{z}|_m^2 - 2\boldsymbol{z}|_m\tilde{\boldsymbol{\mu}}_i|_m + \tilde{\boldsymbol{\mu}}_i|_m^2\right]}{2\tilde{\boldsymbol{\sigma}}_i^2|_m}\right) \\ =& \frac{1}{2}\sum_{m=1}^{M}(-\log\tilde{\boldsymbol{\sigma}}_i^2|_m - 1) \end{aligned} \tag{18}$$

5. Details of the Siamese Network

We train a Siamese network to measure the similarity between the data points. We select N_t nearest neighbors for each datapoint, and group them into N_t pairs. We label these pairs as as positive. We then randomly select N_t pairs of data and treat them as negative. We then minimize the following contrastive loss

$$L = \begin{cases} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 & (\boldsymbol{x}_i, \boldsymbol{x}_j) \text{ is positive pair} \\ \max(c - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2, 0)^2 & (\boldsymbol{x}_i, \boldsymbol{x}_j) \text{ is negative pair} \end{cases} \tag{19}$$

where c is a predefined parameter. In our experiments, the architecture of the Siamese network is set to same with that of encoder, i.e., $D - 500 - 500 - 2000 - 10$. The network is fully connected, and ReLU is used as the activation function. The parameter c is set to 3. N_t is set to 2, 5, 3, 3 for MNIST, STL-10, Reuters, and HHAR, respectively. The network is trained using Adam optimizer with the initial learning rate 0.0005. The learning rate decays every 70 epochs with a factor 0.1. The weight decay in Adam is set to 0.0001

6. JS divergence between two Gaussian

We plot the JS divergence between two Gaussian distribution with varying relative orientation. From the figure, we can find that the JS divergence is minimized when the two Gaussian aligned with the coordinate.

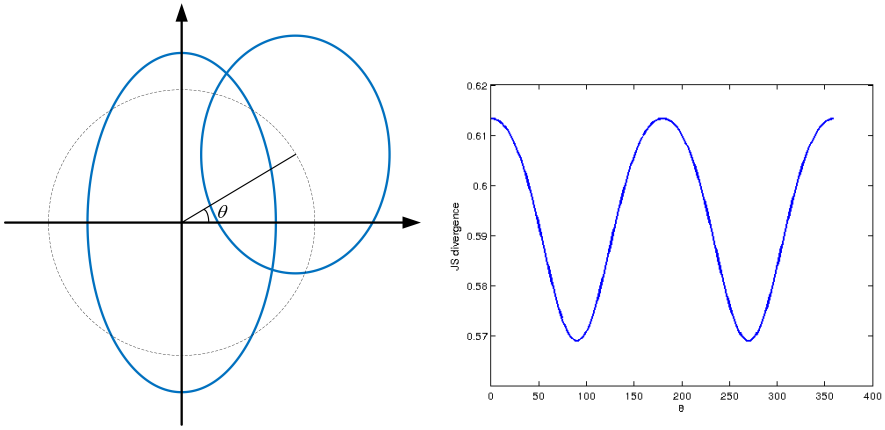

Figure 1. The JS divergence between two Gaussian distributions with different orientation. The two Gaussian distributions have diagonal covariance matrix, and the distance between their mean are fixed.

7. Additional Experimental Analysis

We provide the clustering results of GMM with the latent representation learnt using denoising autoencoder (DAE) on 2-D half circles. Fig.2 shows the learnt representations as well as the clustering results. From the figure we see that the DAE+GMM cannot clusters correctly.

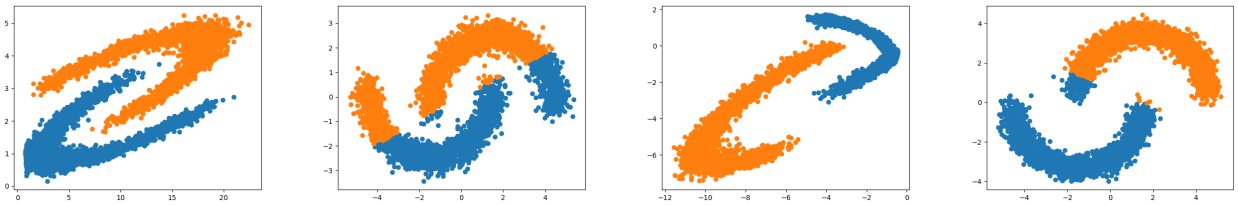

Figure 2. Results of the DAE+GMM on 2D examples with different cluster distance. From left to right: learnt latent features and clustering result for case of small cluster distance, learnt latent features and clustering result for the case of large cluster distance.

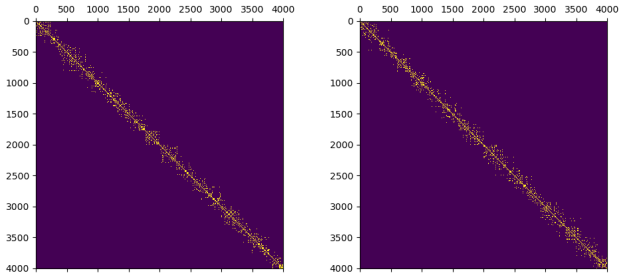

Figure 3. Graphs used in the proposed method on 2D examples. From left to right: graphs for cases of small and large cluster distance, respectively.

We plot the graphs for the proposed method we used to obtain the results on the 2D examples in Fig.3.

We visualize the latent features learnt by the proposed method trained on MNIST at the different training stages. We randomly select 1000 datapoints, and compute their latent features using the learnt encoder. The latent features then are reduced to 2-D using the t-SNE. Fig.4 plots the results using the network before training, after 20 epochs and 100 epochs training, respectively. From Fig.4, we see that after training, the latent features spread more compactly, and the clusters are more pure and well-separated.

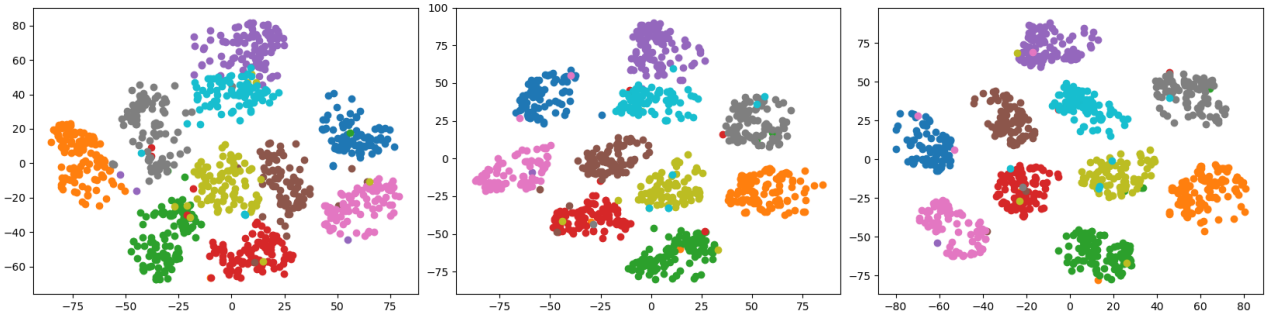

Figure 4. Visualization of the latent features learnt by the proposed method at different training stages. From left to right: latent features obtained using the encoder before training (after pretraining), after 20 epochs and after 300 epochs training, respectively

We discuss the time complexity. In short, compared with VaDE, we have additional overhead of graph embedding, which increases with number of neighbors in the graph. We have measured run time on MNIST for one epoch and show the results in table 1. From the table we see that with 3 neighbors, our method has only slightly higher run time compared to VaDE (our: 8.5s, VaDE: 6.4s), but accuracy is much improved (our: 97.33, VaDE: 94.82)

Table 1. Clustering accuracy and run time for VaDE and the proposed method with different number of neighbors

	VaDE	$N_s = 1$	$N_s = 3$	$N_s = 10$	$N_s = 20$
ACC	94.82	96.98	97.33	97.52	97.58
run time	6.4s	8.7s	8.5s	10.5s	15.5s

8. Images generated by the proposed method

We provides additional images generated by the decoder of our model as well as the learnt variance of the Gaussian components.

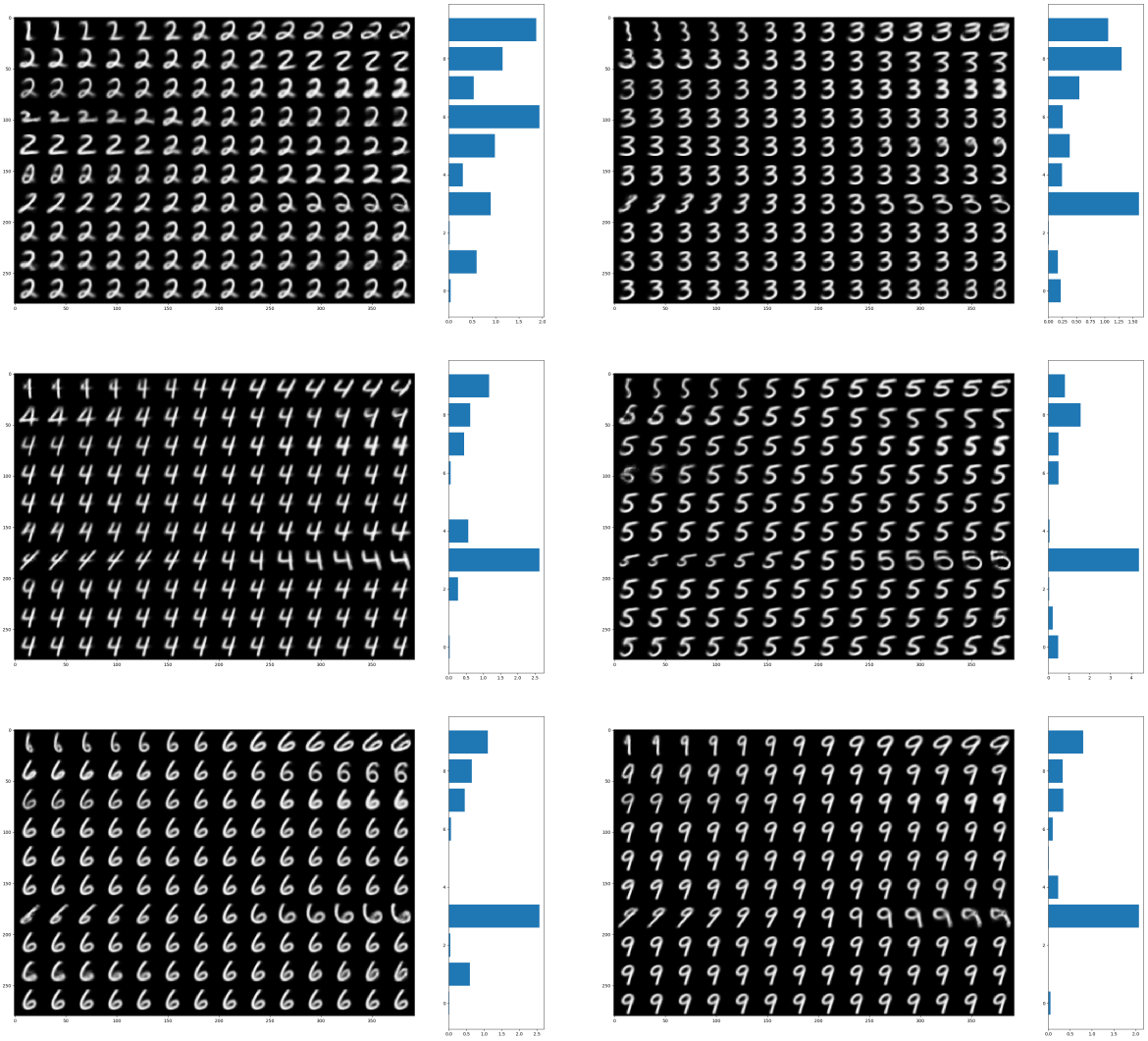

Figure 5. Images generated by the proposed model and estimated variance of the components in GMM