Cluster-wise Hierarchical Generative Model for Deep Amortized Clustering
Supplementary Material

1. Details of ELBO

\[ E_p(x,c_{1:k}) \log p_\theta(C_{1:K}|X) \]
\[ = E_p(x,c_{1:k}) \sum_{k=1}^{K} \log \left( \sum_{L_k=1}^{N_k} \int p_\theta(h^k, z^k, I^k|C_{1:k-1}, S^k) dz^k \right) \]
\[ \geq E_p(x,c_{1:k}) \sum_{k=1}^{K} E_{q_\phi(z^k,I^k|C_{1:k},S^k)} \log \left( \frac{p_\theta(h^k, z^k, I^k|C_{1:k-1}, S^k)}{q_\phi(z^k,I^k|C_{1:k},S^k)} \right) \]
\[ = E_p(x,c_{1:k}) \sum_{k=1}^{K} E_{q_\phi(z^k,I^k|C_{1:k},S^k)} \log \left( \frac{\prod_{i=1}^{M_k-1} p_\theta(h_i^k|z_i^k,S_{I^k})p_\theta(z_{I^k}|S_{I^k})p_\theta(I^k|C_{1:k-1}, S^k)}{q_\phi(z^k|h^k,S_{I^k})q_\phi(I^k|C_{1:k}, S^k)} \right) \]
\[ = \mathcal{L}_{\theta,\phi} \]

2. Implementation Details

2.1. Multi-head attention Module (MhA)

We use Multi-head attention Module MhA(·) to exploit pair-wise or higher-order interactions between data points in both inter-and intra-cluster. Considering we want to capture the elements-wise relationship between A and B, we set A as query, and set key and values are B. The Multi-head attention Module is defined as follow

\[ \text{MhA}(A, B) = \text{concat}(O_1, \cdots, O_H)W^h \]
\[ \text{where} \quad O_i = \sigma \left( A W^Q_i (B O^K_i)^\top \right) BW^V_i \]

where \( \sigma(\cdot) \) is activation function, \( W^Q_i, W^K_i, W^V_i \) are head-specific transform matrices.

2.2. Implementation of \( q_\phi(I^k|C_{1:k}, S^k) \)

For convenience, we draw one-hot vector \( o^k \) from the following categorical distribution:

\[ o^k \sim \text{Categorical} (\text{Softmax}([s_1^k, \cdots, s_j^k, \cdots, s_N^k])) \]
\[ s_j^k = \frac{\text{Cosine}(e_j^k, m^k)}{\tau} - \frac{1}{k-1} \sum_{l=1}^{k-1} \frac{\text{Cosine}(m_l^k, m^k)}{\tau} \]

here \( I^k = \text{index}(\max(o^k)) \)

3. Proof for Theorem 1

Theorem 1 Ergodic amortized inference (EAI) objective \( \mathcal{L}_{\theta,\phi}^* \) serves as a valid lower bound to the log likelihood of data and tighter than the original amortized inference objective \( \mathcal{L}_{\theta,\phi}^* \) and SVI-based amortized inference objective \( \mathcal{L}_{\theta,\phi}^\Delta \). The lower bounds satisfy

\[ \mathcal{L}_{\theta,\phi} \leq \mathcal{L}_{\theta,\phi}^* \leq \mathcal{L}_{\theta,\phi}^\Delta \leq E_p(x,c_{1:k}) \log p_\theta(C_{1:K}|X) \]
Proof. Firstly, we show the following facts about the log-likelihood lower bound $\mathcal{L}_{\theta,\phi}$

$$
\mathcal{L}_{\theta,\phi} = \mathbb{E}_p(x, c_{1:k}) \sum_{k=1}^{K} \mathbb{E} q_0(z^{(k)}_{(0:M)}, t^k | c_{1:k}, s^k) \log \sum_{m=0}^{M} \pi_m u^{(k)}_m \\
\leq \mathbb{E}_p(x, c_{1:k}) \sum_{k=1}^{K} \mathbb{E} \sum_{m=0}^{M} \pi_m q_0(z^{(k)}_{(0:M)}, t^k | c_{1:k}, s^k) u^{(k)}_m \\
= \mathbb{E}_p(x, c_{1:k}) \sum_{k=1}^{K} \sum_{m=0}^{M} \pi_m q_0(z^{(k)}_{(0:M)}, t^k | c_{1:k}, s^k) p_\theta(c_{1:k} | x) \\
= \mathbb{E}_p(x, c_{1:k}) \log p_\theta(c_{1:k} | x)
$$

(5)

Secondly, we prove that $\mathcal{L}_{\theta,\phi} \leq \mathcal{L}_{\theta,\phi}^*$. Let $I \subset \{1, \cdots, M\}$ with $|I| = P$ be a uniformly distributed subset of distinct indices from $\{1, \cdots, M\}$.

$$
\mathcal{L}_{\theta,\phi}^* = \mathbb{E}_p(x, c_{1:k}) \sum_{k=1}^{K} \mathbb{E} q_0(z^{(k)}_{(0:M)}, t^k | c_{1:k}, s^k) \log \sum_{m=0}^{M} \pi_m u^{(k)}_m \\
= \mathbb{E}_p(x, c_{1:k}) \mathbb{E} \sum_{m=0}^{M} \pi_m q_0(z^{(k)}_{(0:M)}, t^k | c_{1:k}, s^k) \log \frac{1}{P} \sum_{i=1}^{P} u^{(k)}_{(m,i)} \\
\geq \mathbb{E}_p(x, c_{1:k}) \mathbb{E} \sum_{m=0}^{M} \pi_m q_0(z^{(k)}_{(0:M)}, t^k | c_{1:k}, s^k) \log \frac{1}{P} \sum_{i=1}^{P} u^{(k)}_{(m,i)} \\
= \mathbb{E}_p(x, c_{1:k}) \mathbb{E} \sum_{m=0}^{M} \pi_m q_0(z^{(k)}_{(0:M)}, t^k | c_{1:k}, s^k) \log \frac{1}{P} \sum_{i=1}^{P} u^{(k)}_{(m,i)}
$$

(6)

Inequality $(a)$ holds due to the Jensen’s inequality, and equality $(b)$ holds since a simple observation:

$$
\mathbb{E}_{I = \{i_1, \cdots, i_P\}} \left[ \frac{a_{i_1} + \cdots + a_{i_P}}{P} \right] = \frac{a_1 + \cdots + a_M}{M}
$$

Based on above inequality, we can derive that

$$
\mathcal{L}_{\theta,\phi}^* \geq \mathbb{E}_p(x, c_{1:k}) \mathbb{E} \sum_{m=0}^{M} \pi_m q_0(z^{(k)}_{(0:M)}, t^k | c_{1:k}, s^k) \log \frac{1}{P} \sum_{i=1}^{P} u^{(k)}_{(m,i)}
$$

(7)

(\text{Eq.})

$$
\mathcal{L}_{\theta,\phi}^* \geq \mathcal{L}_{\theta,\phi}
$$

(8)

and

$$
\mathcal{L}_{\theta,\phi} \geq \mathcal{L}_{\theta,\phi}^* \geq \mathcal{L}_{\theta,\phi}
$$

(9)

Since $z^{(M)}_k$ is the optimal $z_k$, we can obtain

$$
\mathcal{L}_{\theta,\phi} \leq \mathcal{L}_{\theta,\phi}^*
$$

By combining inequalities (5), (7), (8) and (9), we established the bound as stated above.

\[\square\]

3.1. Infrastructure and Experimental Details

Infrastructure: We implement our model with Tensorflow, and conduct our experiments with:
• CPU: Intel Xeon Silver 4116 @2.1GHz.
• GPU: 8x GeForce RTX 2080Ti.
• RAM: DDR4 256GB.
• ROM: 8x 1TB 7.2K 6Gb SATA and 1x 960G SATA 6Gb R SSD
• Operating system: Ubuntu 18.04 LTS.
• Environments: Python 3.7; NumPy 1.18.1; SciPy 1.2.1; scikit-learn 0.23.2; seaborn 0.1; torch_geometric 1.6.1; matplotlib 3.1.3; dgl 0.4.2; pytorch 1.6

**Hyper-parameter search:** We trained with the following hyperparameters: The neural network (e.g., \( f_\theta(\cdot), f_\phi(\cdot), g_{\phi,i}(\cdot) \)) in our model is a multilayer perceptron (MLP). We use the tanh activation function. We apply dropout before every layers, except the last layer. The model is trained using Adam. We then tune the other hyper-parameters of both our approaches and our baselines automatically using the TPE method implemented by Hyeprot. We let Hyperopt conduct 200 trials to search for the optimal hyper-parameter configuration for each method on the validation of each dataset. The hyper-parameter search space is specified as follows:

• The number of hidden layers in a neural network: \{0, 1, 2, 3 \}.
• The number of neurons in a hidden layer: \{100, 200, \cdots, 1000 \}.
• Learning rate: \([10^{-8}, \cdots, 1]\).
• L2 regularization: \([10^{-12}, \cdots, 1]\).
• Dropout rate: [0.05, \cdots, 1].
• Regularization coefficient \(\lambda\): [1, 10].
• The standard deviation of the prior for \(\theta, \phi\): [0.01, 0.5].

**2D MoG data generation:** We generate synthetic data by the following process.

\[
\alpha \sim \text{Exp}(1) \quad c_{1,N} \sim \text{CRP}(\alpha) \\
\mu_k \sim N(0, \sigma_k^2 1) \quad x_i \sim N(\mu_{ci}, \sigma^2 1)
\]

Here we give a more large-size illustration for the generative process, as shown in Figure 1.

**The learned clusters:** In Figure 2 we show top-10 scoring images from each cluster in MNIST and STL-10. Each row corresponds to a cluster and images are sorted from left to right based on the learned \(h_k\). We observe that for MNIST, the cluster assignment corresponds to natural clusters very well, while for STL-10, the results are mostly correct with airplanes, trucks and cars, but spends part of its attention on poses instead of categories when it comes to animal classes.

**References**
Figure 2. Illustration of the learned top-10 images in each cluster of MNIST and STL-10.