

Supplementary Materials

A Collective decision making through multi-arm settings

Proposition 1. Consider the problem of mixture representation learning in a multi-arm VAE framework. For independent samples from category \mathbf{m} , i.e. $\mathbf{x}_i \sim p(\mathbf{x}|\mathbf{m})$,

$$\begin{aligned} \mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{c} = \mathbf{m}|\{\mathbf{x}_i\}_{1:A})] &> \mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{c} = \mathbf{m}|\{\mathbf{x}_i\}_{1:B})] \\ \text{s.t. } \mathbf{c} = \mathbf{c}_1 = \dots = \mathbf{c}_A &\quad \text{s.t. } \mathbf{c} = \mathbf{c}_1 = \dots = \mathbf{c}_B \end{aligned} \quad (1)$$

if $q(\mathbf{m}|\mathbf{x}_i) < 1$ and $A > B \geq 1$ denote the number of arms.

Proof. Given sample \mathbf{x} with categorical variable \mathbf{m} , in a multi-arm framework, each arm receives a noisy copy \mathbf{x}_i with the same categorical factor. Here, \mathbf{c} denotes the joint categorical representation, where $\mathbf{c}_1 = \dots = \mathbf{c}_A = \mathbf{c}$. Defining $\mathbf{X}_A := \{\mathbf{x}_i\}_{1:A}$, for an A -arm VAE, the approximated categorical log posterior can be expressed as,

$$\begin{aligned} \log q(\mathbf{c}_1 = \dots = \mathbf{c}_A = \mathbf{c}|\mathbf{X}_A) &= \log \frac{q(\mathbf{x}_1|\mathbf{X}_A \setminus \{\mathbf{x}_1\}, \mathbf{c})q(\mathbf{x}_2|\mathbf{X}_A \setminus \{\mathbf{x}_1, \mathbf{x}_2\}, \mathbf{c}) \dots q(\mathbf{x}_A|\mathbf{c})q(\mathbf{c})}{q(\mathbf{X}_A)} \\ &= \log \frac{q(\mathbf{x}_1|\mathbf{X}_A \setminus \{\mathbf{x}_1\}, \mathbf{c}) \dots q(\mathbf{x}_A|\mathbf{c})}{q(\mathbf{X}_A)} + \log q(\mathbf{c}) \end{aligned} \quad (2)$$

where we used $q(\mathbf{c}) = q(\mathbf{c}_1 = \dots = \mathbf{c}_A = \mathbf{c})$ to simplify the notation. Since all samples are independently generated, i.e. $q(\mathbf{x}_i|\mathbf{c}, \mathbf{x}_j) = q(\mathbf{x}_i|\mathbf{c})$, the categorical log probability can be simplified as,

$$\begin{aligned} \log q(\mathbf{c}|\mathbf{X}_A) &= \log \prod_{i=1}^A \frac{q(\mathbf{x}_i|\mathbf{c})}{q(\mathbf{x}_i)} + \log q(\mathbf{c}) \\ &= \sum_{i=1}^A \log \frac{q(\mathbf{x}_i|\mathbf{c})}{q(\mathbf{x}_i)} + \log q(\mathbf{c}) \end{aligned} \quad (3)$$

By computing the expectation of the log posterior with respect to the empirical distribution of noisy samples with categorical factor \mathbf{m} , we have,

$$\mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{c}|\mathbf{X}_A)] = \sum_{i=1}^A \mathbb{E}_{q(\mathbf{x}|\mathbf{m})} \left[\log \frac{q(\mathbf{x}_i|\mathbf{c})}{q(\mathbf{x}_i)} \right] + \log q(\mathbf{c}) \quad (4)$$

Since all of the data are independently sampled, the expected log-likelihood values can be expressed as follows.

$$\sum_{i=1}^A \mathbb{E}_{q(\mathbf{x}|\mathbf{c})} [\log q(\mathbf{x}_i|\mathbf{c})] = A \mathbb{E}_{q(\mathbf{x}|\mathbf{c})} [\log q(\mathbf{x}|\mathbf{c})] \quad (5)$$

Therefore, in an A -arm framework, the approximated expected log posterior probability for the joint categorical factor \mathbf{m} is defined as,

$$\mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{c} = \mathbf{m}|\mathbf{X}_A)] = A \mathbb{E}_{q(\mathbf{x}|\mathbf{m})} \left[\log \frac{q(\mathbf{x}|\mathbf{c} = \mathbf{m})}{q(\mathbf{x})} \right] + \log q(\mathbf{c} = \mathbf{m}). \quad (6)$$

According to Eq. 6, for a single arm (1-arm), we have

$$\begin{aligned} \mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{c} = \mathbf{m}|\mathbf{x})] &= \mathbb{E}_{q(\mathbf{x}|\mathbf{m})} \left[\log \frac{q(\mathbf{x}|\mathbf{c} = \mathbf{m})}{q(\mathbf{x})} \right] + \log q(\mathbf{c} = \mathbf{m}), \\ &= D_{KL}(q(\mathbf{x}|\mathbf{c} = \mathbf{m})||q(\mathbf{x})) + \log q(\mathbf{c} = \mathbf{m}) \end{aligned} \quad (7)$$

and for a B -arm framework, we have

$$\mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{c} = \mathbf{m}|\mathbf{X}_B)] = B D_{KL}(q(\mathbf{x}|\mathbf{c} = \mathbf{m})\|q(\mathbf{x})) + \log q(\mathbf{c} = \mathbf{m}). \quad (8)$$

Since $D_{KL}(q(\mathbf{x}|\mathbf{m})\|q(\mathbf{x})) > 0$ (for more than one category), given \mathbf{m} , when for each individual arm $q(\mathbf{c} = \mathbf{m}|\mathbf{x}_i) < 1$ and $A > B \geq 1$, we have

$$A D_{KL}(q(\mathbf{x}|\mathbf{c} = \mathbf{m})\|q(\mathbf{x})) > B D_{KL}(q(\mathbf{x}|\mathbf{c} = \mathbf{m})\|q(\mathbf{x})), \quad (9)$$

$$\mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{c} = \mathbf{m}|\mathbf{x}_1, \dots, \mathbf{x}_A)] > \mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{c} = \mathbf{m}|\mathbf{x}_1, \dots, \mathbf{x}_B)]. \quad (10)$$

□

Proposition 2. *In the A -arm VAE framework, there exists an A that guarantees a true categorical assignment on expectation. That is,*

$$\mathbf{m} = \arg \max_{\mathbf{c}} \mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{c}|\{\mathbf{x}_i\}_{1:A})], \quad \text{s.t. } \mathbf{c} = \mathbf{c}_1 = \dots = \mathbf{c}_A. \quad (11)$$

Proof. In the A -arm framework, an accurate categorical assignment for all samples with the same categorical factor, e.g. \mathbf{m} , can be obtained on expectation, if and only if,

$$\mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{c}_1 = \dots = \mathbf{c}_A = \mathbf{m}|\{\mathbf{x}_i\}_{1:A})] > \mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{c}_1 = \dots = \mathbf{c}_A = \mathbf{n}|\{\mathbf{x}_i\}_{1:A})], \quad \forall \mathbf{n} \neq \mathbf{m},$$

where, here \mathbf{m} is the ground-truth category. In case of the 1-arm framework, the correct categorical assignment receives the highest log posterior probability on expectation, if and only if,

$$\mathbb{E}_{q(\mathbf{x}|\mathbf{m})} \left[\log \frac{q(\mathbf{x}|\mathbf{c} = \mathbf{m})}{q(\mathbf{x}|\mathbf{c} = \mathbf{n})} \right] > \log \frac{q(\mathbf{c} = \mathbf{n})}{q(\mathbf{c} = \mathbf{m})}, \quad \forall \mathbf{n} \neq \mathbf{m} \quad (12)$$

which is a function of categorical distributions and is not always satisfied for any arbitrary discrete distribution. According to Eq. 6, in the presence of A VAE arms, we have

$$\mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{c}_1 = \dots = \mathbf{c}_A = \mathbf{c}|\{\mathbf{x}_i\}_{1:A})] = A \mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{x}|\mathbf{c})] + \log q(\mathbf{c}) - A \mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{x})]. \quad (13)$$

In this framework, the accurate categorical assignment is obtained on expectation, if and only if,

$$\begin{aligned} \mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{c}_1 = \dots = \mathbf{c}_A = \mathbf{m}|\{\mathbf{x}_i\}_{1:A})] &> \mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{c}_1 = \dots = \mathbf{c}_A = \mathbf{n}|\{\mathbf{x}_i\}_{1:A})], \\ A \mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{x}|\mathbf{m})] + \log q(\mathbf{m}) - A \mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{x})] &> A \mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{x}|\mathbf{n})] + \log q(\mathbf{n}) - A \mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{x})], \\ A \mathbb{E}_{q(\mathbf{x}|\mathbf{m})} \left[\log \frac{q(\mathbf{x}|\mathbf{c} = \mathbf{m})}{q(\mathbf{x}|\mathbf{c} = \mathbf{n})} \right] &> \log \frac{q(\mathbf{c} = \mathbf{n})}{q(\mathbf{c} = \mathbf{m})}. \end{aligned} \quad (14)$$

Thus, when the number of arms, A , satisfies

$$A > \max(\rho(\mathbf{m})D^{-1}(\mathbf{m}), 1) \quad (15)$$

where $\rho(\mathbf{m}) = \max_{\mathbf{n} \neq \mathbf{m}} \log \frac{q(\mathbf{c} = \mathbf{n})}{q(\mathbf{c} = \mathbf{m})}$ and $D(\mathbf{m}) = \min_{\mathbf{n} \neq \mathbf{m}} D_{KL}(q(\mathbf{x}|\mathbf{m})\|q(\mathbf{x}|\mathbf{n}))$, maximum of the expected log posterior probability belongs to the true categorical factor. □

Corollary 1. *For a uniform prior on the discrete factor, one pair of VAE arms ($A = 2$) is sufficient to satisfy Eq. 11.*

Proof. For uniformly distributed clusters, $\forall \mathbf{m}$, $\rho(\mathbf{m}) = 0$. According to Eq. 15, for any $A \geq 2$, the accurate categorical assignment is satisfied. □

We further study the under-exploration scenario in data augmentation, in which the noisy samples are concentrated around the given sample. Under this scenario, the proof of earlier Propositions follow in the same way except the augmented samples are no longer conditionally independent. This means that each augmented sample adds less than before to the expected log posterior. Yet, the same argument shows that there will be an A for which the claims of the proposition are satisfied. This issue is discussed in Remark 2 as follows.

Remark 2. When the augmentation is type-preserving, by definition, $q(\mathbf{x}_i|\mathbf{x}_j, \mathbf{c}) = q(\mathbf{x}_i|\mathbf{c})$, where \mathbf{x}_j could be either the given training sample or another noisy copy. If the augmented samples concentrate around \mathbf{x}_j , i.e. the augmenter under-explores the category-conditioned distribution, the earlier proofs should be adapted by keeping the conditioning on \mathbf{x}_j explicit. Conditionally independent terms used in Eq. 2 should be replaced by $q(\mathbf{x}_i|\mathbf{x}_j, \mathbf{c})$ as follows.

$$\log q(\mathbf{c}|\mathbf{X}_A) = \log \frac{q(\mathbf{x}_1|\mathbf{x}_j, \mathbf{X} \setminus \{\mathbf{x}_1, \mathbf{x}_j\}, \mathbf{c}) \dots q(\mathbf{x}_A|\mathbf{x}_j, \mathbf{c})q(\mathbf{x}_j|\mathbf{c})q(\mathbf{c})}{q(\mathbf{X}_A)}. \quad (16)$$

Since all augmented samples are generated from sample \mathbf{x}_j , the conditional probability distribution can be simplified as follows.

$$q(\mathbf{x}_i|\mathbf{x}_j, \mathbf{X}_A \setminus \{\mathbf{x}_i, \mathbf{x}_j\}, \mathbf{c}) = q(\mathbf{x}_i|\mathbf{x}_j, \mathbf{c}), \quad \text{for } i \neq j \quad (17)$$

By computing the expectation of the log posterior with respect to the empirical distribution of noisy samples, $q(\mathbf{X}_A|\mathbf{m})$ (samples are not independent anymore), the expected categorical log posterior probability for the true categorical factor can be defined as,

$$\begin{aligned} \mathbb{E}_{q(\mathbf{X}_A|\mathbf{m})} [\log q(\mathbf{c} = \mathbf{m}|\mathbf{X}_A)] &= \mathbb{E}_{q(\mathbf{x}_i|\mathbf{x}_j, \mathbf{m})} \left[\log \prod_{i=1}^{A-1} \frac{q(\mathbf{x}_i|\mathbf{x}_j, \mathbf{c} = \mathbf{m})}{q(\mathbf{x}_i|\mathbf{x}_j)} \right] + \mathbb{E}_{q(\mathbf{x}_j|\mathbf{m})} \left[\log \frac{q(\mathbf{x}_j|\mathbf{m})}{q(\mathbf{x}_j)} \right] + \log q(\mathbf{c} = \mathbf{m}) \\ &= (A-1)\mathbb{E}_{q(\mathbf{x}_i|\mathbf{x}_j, \mathbf{m})} \left[\log \frac{q(\mathbf{x}_i|\mathbf{x}_j, \mathbf{c} = \mathbf{m})}{q(\mathbf{x}_i|\mathbf{x}_j)} \right] + \mathbb{E}_{q(\mathbf{x}_j|\mathbf{m})} [\log q(\mathbf{c} = \mathbf{m}|\mathbf{x}_j)] \end{aligned} \quad (18)$$

Based on Eq. 18, if the data augmenter only regenerates given sample, the expected log posterior probability in the A -arm framework is equal to the the expected log posterior probability in the single framework, $\mathbb{E}_{q(\mathbf{X}_A|\mathbf{m})} [\log q(\mathbf{c} = \mathbf{m}|\mathbf{X}_A)] = \mathbb{E}_{q(\mathbf{x}|\mathbf{m})} [\log q(\mathbf{c} = \mathbf{m}|\mathbf{x})]$.

B Variational lower bound for conditional single mix-VAE

For completeness, we first derive the evidence lower bound (ELBO) for an observation \mathbf{x} described by one categorical random variable (RV), \mathbf{c} , and one continuous RV, \mathbf{s} , without assuming conditional independence of \mathbf{c} and \mathbf{s} given \mathbf{x} . The variational approach to choosing the latent variables corresponds to solving the optimization equation

$$q^*(\mathbf{s}, \mathbf{c}|\mathbf{x}) = \arg \min_{q(\mathbf{s}, \mathbf{c}|\mathbf{x}) \in \mathcal{D}} D_{\text{KL}}(q(\mathbf{s}, \mathbf{c}|\mathbf{x}) \| p(\mathbf{s}, \mathbf{c}|\mathbf{x})) , \quad (19)$$

where \mathcal{D} is a family of density functions over the latent variables. However, evaluating the objective function requires knowledge of $p(\mathbf{x})$, which is usually unknown. Therefore, we rewrite the divergence term as

$$\begin{aligned}
D_{\text{KL}}(q(\mathbf{s}, \mathbf{c}|\mathbf{x})||p(\mathbf{s}, \mathbf{c}|\mathbf{x})) &= \int_{\mathbf{s}} \sum_{\mathbf{c}} q(\mathbf{s}|\mathbf{c}, \mathbf{x}) q(\mathbf{c}|\mathbf{x}) \log \frac{q(\mathbf{s}|\mathbf{c}, \mathbf{x}) q(\mathbf{c}|\mathbf{x})}{\frac{p(\mathbf{x}|\mathbf{s}, \mathbf{c}) p(\mathbf{s}|\mathbf{c}) p(\mathbf{c})}{p(\mathbf{x})}} d\mathbf{s} \\
&= \int_{\mathbf{s}} \sum_{\mathbf{c}} q(\mathbf{s}|\mathbf{c}, \mathbf{x}) q(\mathbf{c}|\mathbf{x}) \log \frac{q(\mathbf{s}|\mathbf{c}, \mathbf{x})}{p(\mathbf{s}|\mathbf{c})} d\mathbf{s} + \int_{\mathbf{s}} \sum_{\mathbf{c}} q(\mathbf{s}|\mathbf{c}, \mathbf{x}) q(\mathbf{c}|\mathbf{x}) \log \frac{q(\mathbf{c}|\mathbf{x})}{p(\mathbf{c})} d\mathbf{s} \\
&\quad + \int_{\mathbf{s}} \sum_{\mathbf{c}} q(\mathbf{s}|\mathbf{c}, \mathbf{x}) q(\mathbf{c}|\mathbf{x}) \log p(\mathbf{x}) d\mathbf{s} - \int_{\mathbf{s}} \sum_{\mathbf{c}} q(\mathbf{s}|\mathbf{c}, \mathbf{x}) q(\mathbf{c}|\mathbf{x}) \log p(\mathbf{x}|\mathbf{s}, \mathbf{c}) d\mathbf{s} \\
&= \log p(\mathbf{x}) - \mathbb{E}_{q(\mathbf{c}|\mathbf{x})} \left[\mathbb{E}_{q(\mathbf{s}|\mathbf{c}, \mathbf{x})} [\log p(\mathbf{x}|\mathbf{s}, \mathbf{c})] \right] \\
&\quad + \mathbb{E}_{q(\mathbf{c}|\mathbf{x})} [D_{\text{KL}}(q(\mathbf{s}|\mathbf{c}, \mathbf{x})||p(\mathbf{s}|\mathbf{c}))] + \mathbb{E}_{q(\mathbf{s}|\mathbf{c}, \mathbf{x})} [D_{\text{KL}}(q(\mathbf{c}|\mathbf{x})||p(\mathbf{c}))] \quad (20) \\
&= \log p(\mathbf{x}) - \mathcal{L}_{\mathbf{s}} \quad (21)
\end{aligned}$$

Since $\log p(\mathbf{x})$ is not function of the optimization parameters, instead of minimizing Eq. 20, the variational lower bound

$$\mathcal{L}_{\mathbf{s}} = \mathbb{E}_{q(\mathbf{c}|\mathbf{x})} \left[\mathbb{E}_{q(\mathbf{s}|\mathbf{c}, \mathbf{x})} [\log p(\mathbf{x}|\mathbf{s}, \mathbf{c})] \right] - \mathbb{E}_{q(\mathbf{c}|\mathbf{x})} [D_{\text{KL}}(q(\mathbf{s}|\mathbf{c}, \mathbf{x})||p(\mathbf{s}|\mathbf{c}))] - \mathbb{E}_{q(\mathbf{s}|\mathbf{c}, \mathbf{x})} [D_{\text{KL}}(q(\mathbf{c}|\mathbf{x})||p(\mathbf{c}))] \quad (22)$$

can be maximized. We choose $q(\mathbf{s}|\mathbf{c}, \mathbf{x})$ to be a factorized Gaussian, parametrized using the reparametrization trick, and assume that the corresponding prior distribution is also a factorized Gaussian, $\mathbf{s}|\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Similarly, for the categorical variable, we assume a uniform prior, $\mathbf{c} \sim U(K)$.

C Variational inference for multi-arm autoencoding networks

As discussed in the main text, the collective decision making for an A-arm VAE network can be formulated as an equality constrained optimization as follows.

$$\begin{aligned}
\max \quad & \mathcal{L}(\phi_1, \theta_1, \mathbf{x}_1, \mathbf{s}_1, \mathbf{c}_1) + \cdots + \mathcal{L}(\phi_A, \theta_A, \mathbf{x}_A, \mathbf{s}_A, \mathbf{c}_A) \\
\text{s.t.} \quad & \mathbf{c}_1 = \cdots = \mathbf{c}_A
\end{aligned} \quad (23)$$

Without loss of generality, the optimization in Eq. 23 can be rephrased as follows.

$$\begin{aligned}
\max \quad & \mathcal{L}(\phi_1, \theta_1, \mathbf{s}_1, \mathbf{c}_1) + \mathcal{L}(\phi_2, \theta_2, \mathbf{s}_2, \mathbf{c}_2) + \cdots + \mathcal{L}(\phi_A, \theta_A, \mathbf{s}_A, \mathbf{c}_A) \\
\text{s.t.} \quad & \mathbf{c}_1 = \mathbf{c}_2 \\
& \mathbf{c}_1 = \mathbf{c}_3 \\
& \cdots \\
& \mathbf{c}_1 = \mathbf{c}_A \\
& \cdots \\
& \mathbf{c}_{A-1} = \mathbf{c}_A
\end{aligned} \quad (24)$$

where the equality constraint is represented as $\binom{A}{2}$ pairs of categorical agreements. Multiplying the objective term in Eq. 23 by a constant value, $A - 1$, we obtain,

$$\begin{aligned}
\max \quad & (A - 1) (\mathcal{L}(\phi_1, \theta_1, \mathbf{s}_1, \mathbf{c}_1) + \mathcal{L}(\phi_2, \theta_2, \mathbf{s}_2, \mathbf{c}_2) + \cdots + \mathcal{L}(\phi_A, \theta_A, \mathbf{s}_A, \mathbf{c}_A)) \\
\text{s.t.} \quad & \mathbf{c}_a = \mathbf{c}_b \quad \forall a, b \in [1, A], a < b
\end{aligned} \quad (25)$$

Consider one pair of \mathcal{L} objectives for two arms a and b :

$$\begin{aligned} \mathcal{L}(\phi_a, \theta_a, \mathbf{s}_a, \mathbf{c}_a) + \mathcal{L}(\phi_b, \theta_b, \mathbf{s}_b, \mathbf{c}_b) &= \mathbb{E}_{q_{\phi_a}(\mathbf{s}_a, \mathbf{c}_a | \mathbf{x}_a)} [\log p_{\theta_a}(\mathbf{x}_a | \mathbf{s}_a, \mathbf{c}_a)] + \mathbb{E}_{q_{\phi_b}(\mathbf{s}_b, \mathbf{c}_b | \mathbf{x}_b)} [\log p_{\theta_b}(\mathbf{x}_b | \mathbf{s}_b, \mathbf{c}_b)] \\ &\quad - \mathbb{E}_{q_{\phi_a}(\mathbf{c}_a | \mathbf{x}_a)} \left[D_{KL}(q_{\phi_a}(\mathbf{s}_a | \mathbf{c}_a, \mathbf{x}_a) \| p(\mathbf{s}_a | \mathbf{c}_a)) \right] - \mathbb{E}_{q_{\phi_b}(\mathbf{c}_b | \mathbf{x}_b)} \left[D_{KL}(q_{\phi_b}(\mathbf{s}_b | \mathbf{c}_b, \mathbf{x}_b) \| p(\mathbf{s}_b | \mathbf{c}_b)) \right] \\ &\quad - \mathbb{E}_{q_{\phi_a}(\mathbf{s}_a | \mathbf{c}_a, \mathbf{x}_a)} \left[D_{KL}(q_{\phi_a}(\mathbf{c}_a | \mathbf{x}_a) \| p(\mathbf{c}_a)) \right] - \mathbb{E}_{q_{\phi_b}(\mathbf{s}_b | \mathbf{c}_b, \mathbf{x}_b)} \left[D_{KL}(q_{\phi_b}(\mathbf{c}_b | \mathbf{x}_b) \| p(\mathbf{c}_b)) \right] \end{aligned} \quad (26)$$

Since all arms receive augmented samples from the same original distribution, we have $p(\mathbf{c}_a) = p(\mathbf{c}_b) = p(\mathbf{c})$. Using a simplified notation, $q_a = q_{\phi_a}(\mathbf{c}_a | \mathbf{x}_a)$, the last two KL divergence terms can be expressed as,

$$\begin{aligned} D_{KL}(q_a \| p(\mathbf{c})) + D_{KL}(q_b \| p(\mathbf{c})) &= \sum_{\mathbf{c}_a} q_a \log \frac{q_a}{p(\mathbf{c})} + \sum_{\mathbf{c}_b} q_b \log \frac{q_b}{p(\mathbf{c})} \\ &= \sum_{\mathbf{c}_a} \sum_{\mathbf{c}_b} q_a q_b \log \frac{q_a}{p(\mathbf{c})} + \sum_{\mathbf{c}_a} \sum_{\mathbf{c}_b} q_a q_b \log \frac{q_b}{p(\mathbf{c})} \\ &= \sum_{\mathbf{c}_a} \sum_{\mathbf{c}_b} q_a q_b \log \frac{q_a q_b}{p(\mathbf{c})} \end{aligned} \quad (27)$$

Now, if we marginalize $p(\mathbf{c})$ over the joint distribution $p(\mathbf{c}_a, \mathbf{c}_b)$, we can represent the categorical prior distribution as follows.

$$p(\mathbf{c}) = \sum_{\mathbf{c}_a, \mathbf{c}_b} p(\mathbf{c} | \mathbf{c}_a, \mathbf{c}_b) p(\mathbf{c}_a, \mathbf{c}_b) \quad (28)$$

Since there is a categorical agreement condition i.e., $\mathbf{c}_a = \mathbf{c}_b$, $p(\mathbf{c})$ can be expressed as,

$$p(\mathbf{c}) = \sum_{\mathbf{m}} p(\mathbf{c} | \mathbf{c}_a = \mathbf{c}_b = \mathbf{m}) p(\mathbf{c}_a = \mathbf{c}_b = \mathbf{m}) \quad (29)$$

where

$$p(\mathbf{c} | \mathbf{c}_a = \mathbf{c}_b = \mathbf{m}) = \begin{cases} 1 & \mathbf{m} = \mathbf{c} \\ 0 & \text{otherwise} \end{cases} \quad (30)$$

Accordingly, under the $\mathbf{c}_a = \mathbf{c}_b$ constraint, we merge those KL divergence terms as follows:

$$\begin{aligned} D_{KL}(q_a \| p(\mathbf{c})) + D_{KL}(q_b \| p(\mathbf{c})) &= \sum_{\mathbf{c}_a} \sum_{\mathbf{c}_b} q_a q_b \log \frac{q_a q_b}{p(\mathbf{c}_a, \mathbf{c}_b)} \\ &= D_{KL}(q_a q_b \| p(\mathbf{c}_a, \mathbf{c}_b)) \end{aligned} \quad (31)$$

Finally, the optimization in Eq. 25 can be expressed as

$$\begin{aligned} \max \quad & \sum_{a=1}^A (A-1) \left(\mathbb{E}_{q(\mathbf{s}_a, \mathbf{c}_a | \mathbf{x}_a)} [\log p(\mathbf{x}_a | \mathbf{s}_a, \mathbf{c}_a)] - \mathbb{E}_{q(\mathbf{c}_a | \mathbf{x}_a)} \left[D_{KL}(q(\mathbf{s}_a | \mathbf{c}_a, \mathbf{x}_a) \| p(\mathbf{s}_a | \mathbf{c}_a)) \right] \right) - \\ & \sum_{a < b} \mathbb{E}_{q(\mathbf{s}_a, \mathbf{s}_b | \mathbf{c}_a, \mathbf{c}_b, \mathbf{x}_a, \mathbf{x}_b)} \left[D_{KL}(q(\mathbf{c}_a | \mathbf{x}_a) q(\mathbf{c}_b | \mathbf{x}_b) \| p(\mathbf{c}_a, \mathbf{c}_b)) \right] \\ \text{s.t.} \quad & \mathbf{c}_a = \mathbf{c}_b \quad \forall a, b \in [1, A], a < b \end{aligned} \quad (32)$$

D Variational lower bound for cpl-mixVAE

In this section, using a pair of VAE arms, first we generalize the loss function for the single mix-VAE i.e., \mathcal{L}_s in Eq. 22, to the multi-arm case, and show that we can achieve the same objective function in Eq. 32. Then, we derive a relaxation for the equality constrained optimization.

Given input data \mathbf{x}_a , an arm approximates two models $q(\mathbf{c}_a|\mathbf{x}_a)$ and $q(\mathbf{s}_a|\mathbf{x}_a, \mathbf{c}_a)$. If we use pairwise coupling to allow interactions between the arms, then, for a pair of VAE arms, a and b , the variational lower bound obtained from the KL divergence in Equation (20) can be generalized as

$$\begin{aligned}\Delta(a, b) &\triangleq D_{\text{KL}}(q(\mathbf{s}_a, \mathbf{s}_b, \mathbf{c}_a, \mathbf{c}_b|\mathbf{x}_a, \mathbf{x}_b) \| p(\mathbf{s}_a, \mathbf{s}_b, \mathbf{c}_a, \mathbf{c}_b|\mathbf{x}_a, \mathbf{x}_b)) \\ &= \int_{\mathbf{s}_a} \int_{\mathbf{s}_b} \sum_{\mathbf{c}_a} \sum_{\mathbf{c}_b} q(\mathbf{s}_a, \mathbf{s}_b|\mathbf{c}_a, \mathbf{c}_b, \mathbf{x}_a, \mathbf{x}_b) q(\mathbf{c}_a, \mathbf{c}_b|\mathbf{x}_a, \mathbf{x}_b) \\ &\quad \times \log \frac{q(\mathbf{s}_a, \mathbf{s}_b|\mathbf{c}_a, \mathbf{c}_b, \mathbf{x}_a, \mathbf{x}_b) q(\mathbf{c}_a, \mathbf{c}_b|\mathbf{x}_a, \mathbf{x}_b)}{\left(\frac{p(\mathbf{x}_a, \mathbf{x}_b|\mathbf{s}_a, \mathbf{s}_b, \mathbf{c}_a, \mathbf{c}_b) p(\mathbf{s}_a, \mathbf{s}_b|\mathbf{c}_a, \mathbf{c}_b) p(\mathbf{c}_a, \mathbf{c}_b)}{p(\mathbf{x}_a, \mathbf{x}_b)} \right)} d\mathbf{s}_a d\mathbf{s}_b \quad (33)\end{aligned}$$

When each arm learns the continuous factor independent of other arms, we have $q(\mathbf{s}_a, \mathbf{s}_b|\mathbf{c}_a, \mathbf{c}_b, \mathbf{x}_a, \mathbf{x}_b) = q(\mathbf{s}_a|\mathbf{c}_a, \mathbf{x}_a) q(\mathbf{s}_b|\mathbf{c}_b, \mathbf{x}_b)$. Equivalently, for independent samples \mathbf{x}_a and \mathbf{x}_b , we have $q(\mathbf{c}_a, \mathbf{c}_b|\mathbf{x}_a, \mathbf{x}_b) = q(\mathbf{c}_a|\mathbf{x}_a) q(\mathbf{c}_b|\mathbf{x}_b)$. Hence,

$$\begin{aligned}\Delta(a, b) &= \log p(\mathbf{x}_a, \mathbf{x}_b) + \int_{\mathbf{s}_a} \int_{\mathbf{s}_b} \sum_{\mathbf{c}_a} \sum_{\mathbf{c}_b} q(\mathbf{s}_a|\mathbf{c}_a, \mathbf{x}_a) q(\mathbf{s}_b|\mathbf{c}_b, \mathbf{x}_b) q(\mathbf{c}_a|\mathbf{x}_a) q(\mathbf{c}_b|\mathbf{x}_b) \log \frac{q(\mathbf{c}_a|\mathbf{x}_a) q(\mathbf{c}_b|\mathbf{x}_b)}{p(\mathbf{c}_a, \mathbf{c}_b)} d\mathbf{s}_a d\mathbf{s}_b \\ &\quad + \int_{\mathbf{s}_a} \sum_{\mathbf{c}_a} q(\mathbf{s}_a|\mathbf{c}_a, \mathbf{x}_a) q(\mathbf{c}_a|\mathbf{x}_a) \log \frac{q(\mathbf{s}_a|\mathbf{c}_a, \mathbf{x}_a)}{p(\mathbf{s}_a|\mathbf{c}_a)} d\mathbf{s}_a + \int_{\mathbf{s}_b} \sum_{\mathbf{c}_b} q(\mathbf{s}_b|\mathbf{c}_b, \mathbf{x}_b) q(\mathbf{c}_b|\mathbf{x}_b) \log \frac{q(\mathbf{s}_b|\mathbf{c}_b, \mathbf{x}_b)}{p(\mathbf{s}_b|\mathbf{c}_b)} d\mathbf{s}_b \\ &\quad - \int_{\mathbf{s}_a} \sum_{\mathbf{c}_a} q(\mathbf{s}_a|\mathbf{c}_a, \mathbf{x}_a) q(\mathbf{c}_a|\mathbf{x}_a) \log p(\mathbf{x}_a|\mathbf{s}_a, \mathbf{c}_a) d\mathbf{s}_a - \int_{\mathbf{s}_b} \sum_{\mathbf{c}_b} q(\mathbf{s}_b|\mathbf{c}_b, \mathbf{x}_b) q(\mathbf{c}_b|\mathbf{x}_b) \log p(\mathbf{x}_b|\mathbf{s}_b, \mathbf{c}_b) d\mathbf{s}_b \quad (34)\end{aligned}$$

$$\begin{aligned}\Delta(a, b) &= -\mathbb{E}_{q(\mathbf{c}_a|\mathbf{x}_a)} \left[\mathbb{E}_{q(\mathbf{s}_a|\mathbf{c}_a, \mathbf{x}_a)} [\log p(\mathbf{x}_a|\mathbf{s}_a, \mathbf{c}_a)] \right] - \mathbb{E}_{q(\mathbf{c}_b|\mathbf{x}_b)} \left[\mathbb{E}_{q(\mathbf{s}_b|\mathbf{c}_b, \mathbf{x}_b)} [\log p(\mathbf{x}_b|\mathbf{s}_b, \mathbf{c}_b)] \right] \\ &\quad + \mathbb{E}_{q(\mathbf{c}_a|\mathbf{x}_a)} \left[D_{\text{KL}}(q(\mathbf{s}_a|\mathbf{c}_a, \mathbf{x}_a) \| p(\mathbf{s}_a|\mathbf{c}_a)) \right] + \mathbb{E}_{q(\mathbf{c}_b|\mathbf{x}_b)} \left[D_{\text{KL}}(q(\mathbf{s}_b|\mathbf{c}_b, \mathbf{x}_b) \| p(\mathbf{s}_b|\mathbf{c}_b)) \right] \\ &\quad + \mathbb{E}_{q(\mathbf{s}_a|\mathbf{c}_a, \mathbf{x}_a)} \left[\mathbb{E}_{q(\mathbf{s}_b|\mathbf{c}_b, \mathbf{x}_b)} \left[D_{\text{KL}}(q(\mathbf{c}_a|\mathbf{x}_a) q(\mathbf{c}_b|\mathbf{x}_b) \| p(\mathbf{c}_a, \mathbf{c}_b)) \right] \right] + \log p(\mathbf{x}_a, \mathbf{x}_b) \quad (35)\end{aligned}$$

Therefore, the variational lower bound for a pair of coupled VAE arms can be expressed as,

$$\begin{aligned}\mathcal{L}_{\text{pair}}(a, b) &= \mathbb{E}_{q(\mathbf{s}_a, \mathbf{c}_a|\mathbf{x}_a)} [\log p(\mathbf{x}_a|\mathbf{s}_a, \mathbf{c}_a)] + \mathbb{E}_{q(\mathbf{s}_b, \mathbf{c}_b|\mathbf{x}_b)} [\log p(\mathbf{x}_b|\mathbf{s}_b, \mathbf{c}_b)] \\ &\quad - \mathbb{E}_{q(\mathbf{c}_a|\mathbf{x}_a)} \left[D_{\text{KL}}(q(\mathbf{s}_a|\mathbf{c}_a, \mathbf{x}_a) \| p(\mathbf{s}_a|\mathbf{c}_a)) \right] - \mathbb{E}_{q(\mathbf{c}_b|\mathbf{x}_b)} \left[D_{\text{KL}}(q(\mathbf{s}_b|\mathbf{c}_b, \mathbf{x}_b) \| p(\mathbf{s}_b|\mathbf{c}_b)) \right] \\ &\quad - \mathbb{E}_{q(\mathbf{s}_a|\mathbf{c}_a, \mathbf{x}_a)} \left[\mathbb{E}_{q(\mathbf{s}_b|\mathbf{c}_b, \mathbf{x}_b)} \left[D_{\text{KL}}(q(\mathbf{c}_a|\mathbf{x}_a) q(\mathbf{c}_b|\mathbf{x}_b) \| p(\mathbf{c}_a, \mathbf{c}_b)) \right] \right] \quad (36)\end{aligned}$$

which is equivalent to the loss function in Eq. 32, for $A = 2$.

To compute the joint distribution $p(\mathbf{c}_a, \mathbf{c}_b)$, here, we define an auxiliary continuous random variable e representing the mismatch (error) between \mathbf{c}_a and \mathbf{c}_b such that $\forall \mathbf{c}_a, \mathbf{c}_b \in \mathcal{S}^K$, and $0 < \epsilon \ll 1$,

$$p(\mathbf{c}_a, \mathbf{c}_b|e) = \begin{cases} 1 & |e - d^2(\mathbf{c}_a, \mathbf{c}_b)| < \epsilon/2 \\ 0 & \text{otherwise} \end{cases} \quad (37)$$

Here, $d(\mathbf{c}_a, \mathbf{c}_b)$ denotes the distance between \mathbf{c}_a and \mathbf{c}_b in the simplex \mathcal{S}^K , as a measure of mismatch between categorical variables. The random variable e is distributed according to an exponential probability density function with parameter λ i.e., $\forall e \geq 0$, $f(e, \lambda) = \lambda \exp(-\lambda e)$, where $\lambda > 0$. Accordingly, the joint categorical distribution can be represented as,

$$p(\mathbf{c}_a, \mathbf{c}_b) = \int p(\mathbf{c}_a, \mathbf{c}_b|e) p(e) de \quad (38)$$

$$= \int_{-\epsilon/2 + d^2(\mathbf{c}_a, \mathbf{c}_b)}^{\epsilon/2 + d^2(\mathbf{c}_a, \mathbf{c}_b)} f(e, \lambda) de = \epsilon f(d^2(\mathbf{c}_a, \mathbf{c}_b), \lambda) + E \quad (39)$$

where E is the error bound of the Midpoint integral rule. For given exponential function $f(e, \lambda)$, since $|f''(e, \lambda)| \leq \lambda^3$, $\forall e > 0$, the Midpoint approximation error is bounded by,

$$|E| \leq \frac{(\lambda\epsilon)^3}{24}. \quad (40)$$

Subsequently, the joint probability distribution is equivalent to:

$$p(\mathbf{c}_a, \mathbf{c}_b) = \epsilon\lambda \exp\left(-\lambda d^2(\mathbf{c}_a, \mathbf{c}_b)\right) + E \quad (41)$$

where ϵ and λ are arbitrarily constant values. We can approximate the joint distribution as follows.

$$p(\mathbf{c}_a, \mathbf{c}_b) \approx \epsilon\lambda \exp\left(-\lambda d^2(\mathbf{c}_a, \mathbf{c}_b)\right) \quad (42)$$

Thus, the last KL divergence in Eq. 36 can be approximated as,

$$\begin{aligned} D_{\text{KL}}(q(\mathbf{c}_a|\mathbf{x}_a)q(\mathbf{c}_b|\mathbf{x}_b)||p(\mathbf{c}_a, \mathbf{c}_b)) &= \sum_{\mathbf{c}_a} \sum_{\mathbf{c}_b} q(\mathbf{c}_a|\mathbf{x}_a)q(\mathbf{c}_b|\mathbf{x}_b) \log \frac{q(\mathbf{c}_a|\mathbf{x}_a)q(\mathbf{c}_b|\mathbf{x}_b)}{p(\mathbf{c}_a, \mathbf{c}_b)} \\ &= -H(\mathbf{c}_a|\mathbf{x}_a) - H(\mathbf{c}_b|\mathbf{x}_b) - \sum_{\mathbf{c}_a} \sum_{\mathbf{c}_b} q(\mathbf{c}_a|\mathbf{x}_a)q(\mathbf{c}_b|\mathbf{x}_b) \log p(\mathbf{c}_a, \mathbf{c}_b) \\ &\approx -H(\mathbf{c}_a|\mathbf{x}_a) - H(\mathbf{c}_b|\mathbf{x}_b) + \lambda \mathbb{E}_{q(\mathbf{c}_a|\mathbf{x}_a)} \mathbb{E}_{q(\mathbf{c}_b|\mathbf{x}_b)} \left[d^2(\mathbf{c}_a, \mathbf{c}_b) \right] - \log \epsilon\lambda, \end{aligned} \quad (43)$$

Therefore, the approximated variational cost for a pair of VAE arms can be written as follows:

$$\begin{aligned} \mathcal{L}_{\text{pair}}(a, b) &= \mathbb{E}_{q(\mathbf{s}_a, \mathbf{c}_a|\mathbf{x}_a)} [\log p(\mathbf{x}_a|\mathbf{s}_a, \mathbf{c}_a)] + \mathbb{E}_{q(\mathbf{s}_b, \mathbf{c}_b|\mathbf{x}_b)} [\log p(\mathbf{x}_b|\mathbf{s}_b, \mathbf{c}_b)] \\ &\quad - \mathbb{E}_{q(\mathbf{c}_a|\mathbf{x}_a)} \left[D_{\text{KL}}(q(\mathbf{s}_a|\mathbf{c}_a, \mathbf{x}_a)||p(\mathbf{s}_a|\mathbf{c}_a)) \right] - \mathbb{E}_{q(\mathbf{c}_b|\mathbf{x}_b)} \left[D_{\text{KL}}(q(\mathbf{s}_b|\mathbf{c}_b, \mathbf{x}_b)||p(\mathbf{s}_b|\mathbf{c}_b)) \right] \\ &\quad + H(\mathbf{c}_a|\mathbf{x}_a) + H(\mathbf{c}_b|\mathbf{x}_b) - \lambda \mathbb{E}_{q(\mathbf{c}_a|\mathbf{x}_a)} \mathbb{E}_{q(\mathbf{c}_b|\mathbf{x}_b)} \left[d^2(\mathbf{c}_a, \mathbf{c}_b) \right] \end{aligned} \quad (45)$$

Now, by extending $\mathcal{L}_{\text{pair}}$ from two arms to A arms, in which there are $\binom{A}{2}$ paired networks, the total loss function for A arms can be written as

$$\begin{aligned} \mathcal{L}_{\text{cpl}} &= \sum_{a=1}^{A-1} \sum_{b=a+1}^A \mathcal{L}_{\text{pair}}(a, b) \\ &= \sum_{a=1}^A (A-1) \mathbb{E}_{q(\mathbf{s}_a, \mathbf{c}_a|\mathbf{x}_a)} [\log p(\mathbf{x}_a|\mathbf{s}_a, \mathbf{c}_a)] - (A-1) \mathbb{E}_{q(\mathbf{c}_a|\mathbf{x}_a)} \left[D_{\text{KL}}(q(\mathbf{s}_a|\mathbf{c}_a, \mathbf{x}_a)||p(\mathbf{s}_a|\mathbf{c}_a)) \right] \\ &\quad + \sum_{a < b} H(\mathbf{c}_a|\mathbf{x}_a) + H(\mathbf{c}_b|\mathbf{x}_b) - \lambda \mathbb{E}_{q(\mathbf{c}_a|\mathbf{x}_a)} \mathbb{E}_{q(\mathbf{c}_b|\mathbf{x}_b)} \left[d^2(\mathbf{c}_a, \mathbf{c}_b) \right]. \end{aligned} \quad (46)$$

E Proof of Proposition 3

E.1 Aitchison geometry

In this section, we first briefly review some critical definitions in *Aitchison geometry*. Then, to support the proof of Proposition 3, here we introduce Lemma 1 and Propositions 4 and 5.

According to Aitchison geometry, a simplex of K parts can be considered as a vector space $(\mathcal{S}^K, \oplus, \otimes)$, in which \oplus and \otimes corresponds to *perturbation* and *power* operations, respectively, as follows.

$$\text{Perturbation} : \forall \mathbf{x}, \mathbf{y} \in \mathcal{S}^K, \mathbf{x} \oplus \mathbf{y} = \mathcal{C}(x_1 y_1, \dots, x_K y_K)$$

$$\text{Power} : \forall \mathbf{x} \in \mathcal{S}^K \text{ and } \forall \alpha \in \mathbb{R}, \alpha \otimes \mathbf{x} = \mathcal{C}(x_1^\alpha, \dots, x_K^\alpha)$$

where \mathcal{C} denotes the closure operation as follows.

$$\mathcal{C}(\mathbf{x}) = \left(\frac{\frac{cx_1}{K}}{\sum_{k=1}^K x_k}, \dots, \frac{\frac{cx_K}{K}}{\sum_{k=1}^K x_k} \right).$$

In the simplex vector space, for any $\mathbf{x}, \mathbf{y} \in \mathcal{S}^K$, the distance is defined as,

$$d_{\mathcal{S}^K}(\mathbf{x}, \mathbf{y}) = \left(\frac{1}{K} \sum_{i < j} \left(\log \frac{x_i}{x_j} - \log \frac{y_i}{y_j} \right)^2 \right)^{1/2}. \quad (47)$$

Furthermore, Aitchison has introduced *centered-logratio* transformation (CLR), which is an isometric transformation from a simplex to a K -dimensional real space, $clr(\mathbf{x}) \in \mathbb{R}^K$. The CLR transformation involves the logratio of each x_k over geometric means in the simplex as follows.

$$clr(\mathbf{x}) = \left(\log \frac{x_1}{g(\mathbf{x})}, \dots, \log \frac{x_K}{g(\mathbf{x})} \right). \quad (48)$$

where $g(\mathbf{x}) = \left(\prod_{k=1}^K x_k \right)^{1/K}$ and $\sum_{k=1}^K \log \frac{x_k}{g(\mathbf{x})} = 0$.

Since CLR is an isometric transformation, we have

$$\begin{aligned} d_{\mathcal{S}^K}(\mathbf{x}, \mathbf{y}) &= d_{\mathbb{R}^K}(clr(\mathbf{x}), clr(\mathbf{y})) \\ &= \|clr(\mathbf{x}) - clr(\mathbf{y})\|_2. \end{aligned}$$

The algebraic-geometric definition of \mathcal{S}^K satisfies standard properties, such as

$$d_{\mathcal{S}^K}(\mathbf{x} \oplus \mathbf{v}, \mathbf{y} \oplus \mathbf{v}) = d_{\mathcal{S}^K}(\mathbf{x} \ominus \mathbf{v}, \mathbf{y} \ominus \mathbf{v}) = d_{\mathcal{S}^K}(\mathbf{x}, \mathbf{y}) \quad (49)$$

where $\mathbf{v} \in \mathcal{S}^K$ could be any arbitrary vector in the simplex.

Lemma 1. Given a set of vectors $\{\mathbf{x}_1, \dots, \mathbf{x}_N\} \in \mathcal{S}^K$ where \mathcal{S}^K is a simplex of K parts, then

$$clr(\mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \dots \oplus \mathbf{x}_N) = clr(\mathbf{x}_1) + clr(\mathbf{x}_2) + \dots + clr(\mathbf{x}_N).$$

Proof. According to Aitchison geometry, addition of vectors in the simplex is defined as,

$$\mathbf{x}_1 \oplus \dots \oplus \mathbf{x}_N = \left(\frac{\prod_{n=1}^N x_{n1}}{c_N}, \dots, \frac{\prod_{n=1}^N x_{nK}}{c_N} \right) \quad (50)$$

where $c_N = \sum_{k=1}^K \prod_{n=1}^N x_{nk}$.

By applying the centered-logratio transformation, we have

$$clr(\mathbf{x}_1 \oplus \dots \oplus \mathbf{x}_N) = \left(\log \frac{\prod_{n=1}^N x_{n1}}{\delta_{K,N}}, \dots, \log \frac{\prod_{n=1}^N x_{nK}}{\delta_{K,N}} \right) \quad (51)$$

where $\delta_{K,N} = c_N \left(\frac{\prod_{k=1}^K \prod_{n=1}^N x_{n_k}}{c_N} \right)^{1/K} = \left(\prod_{k=1}^K \prod_{n=1}^N x_{n_k} \right)^{1/K}$.

Now, we can rewrite Eq. 51 as,

$$\begin{aligned} clr(\mathbf{x}_1 \oplus \dots \oplus \mathbf{x}_N) &= \left(\log \frac{x_{1_1} \dots x_{N_1}}{\left(\prod_k x_{1_k} \right)^{1/K} \dots \left(\prod_k x_{N_k} \right)^{1/K}}, \dots, \log \frac{x_{1_K} \dots x_{N_K}}{\left(\prod_k x_{1_k} \right)^{1/K} \dots \left(\prod_k x_{N_k} \right)^{1/K}} \right) \\ &= \left(\sum_n \log \frac{x_{n_1}}{\left(\prod_k x_{n_k} \right)^{1/K}}, \dots, \sum_n \log \frac{x_{n_K}}{\left(\prod_k x_{n_k} \right)^{1/K}} \right) \\ &= clr(x_1) + \dots + clr(x_N) \end{aligned} \tag{52}$$

□

Proposition 4. Given vectors $\mathbf{x}, \mathbf{y}, \mathbf{v}_x, \mathbf{v}_y \in \mathcal{S}^K$ where \mathcal{S}^K is a simplex of $K > 0$ parts, then

$$d_{\mathcal{S}^K}^2(\mathbf{x}, \mathbf{y}) - \Gamma_l \leq d_{\mathcal{S}^K}^2(\mathbf{x} \oplus \mathbf{v}_x, \mathbf{y} \oplus \mathbf{v}_y) \leq d_{\mathcal{S}^K}^2(\mathbf{x}, \mathbf{y}) + \Gamma_u$$

where $\Gamma_u, \Gamma_l \geq 0$, $\Gamma_u = K\tau_u^2 - \frac{\Delta^2}{K}$, $\Gamma_l = \frac{\Delta^2}{K} - K\tau_l^2$, $\tau_u = \max_k \{\log \frac{v_{x_k}}{v_{y_k}}\}$, $\tau_l = \max_k \{\log \frac{v_{y_k}}{v_{x_k}}\}$, and $\Delta = \sum_k \left(\log \frac{v_{x_k}}{v_{y_k}} \right)$.

Proof. According to Aitchison geometry, the distance between two vectors $\mathbf{x}, \mathbf{y} \in \mathcal{S}^K$ is defined as,

$$d_{\mathcal{S}^K}^2(\mathbf{x}, \mathbf{y}) = \|clr(\mathbf{x}) - clr(\mathbf{y})\|_2^2$$

If we perturb vectors \mathbf{x} and \mathbf{y} by \mathbf{v}_x and \mathbf{v}_y , the distance between the perturbed vectors in the simple can be expressed as,

$$d_{\mathcal{S}^K}^2(\mathbf{x} \oplus \mathbf{v}_x, \mathbf{y} \oplus \mathbf{v}_y) = \|clr(\mathbf{x} \oplus \mathbf{v}_x) - clr(\mathbf{y} \oplus \mathbf{v}_y)\|_2^2.$$

According to Lemma 1,

$$\begin{aligned} d_{\mathcal{S}^K}^2(\mathbf{x} \oplus \mathbf{v}_x, \mathbf{y} \oplus \mathbf{v}_y) &= \| (clr(\mathbf{x}) - clr(\mathbf{y})) + (clr(\mathbf{v}_x) - clr(\mathbf{v}_y)) \|_2^2 \\ &= \|clr(\mathbf{x}) - clr(\mathbf{y})\|_2^2 + \|clr(\mathbf{v}_x) - clr(\mathbf{v}_y)\|_2^2 + \\ &\quad (clr(\mathbf{x}) - clr(\mathbf{y}))^T (clr(\mathbf{v}_x) - clr(\mathbf{v}_y)) + (clr(\mathbf{v}_x) - clr(\mathbf{v}_y))^T (clr(\mathbf{x}) - clr(\mathbf{y})) \\ &= d_{\mathcal{S}^K}^2(\mathbf{x}, \mathbf{y}) + d_{\mathcal{S}^K}^2(\mathbf{v}_x, \mathbf{v}_y) + 2 \sum_{k=1}^K \left(\log \frac{x_k}{g(\mathbf{x})} - \log \frac{y_k}{g(\mathbf{y})} \right) \left(\log \frac{v_{x_k}}{g(\mathbf{v}_x)} - \log \frac{v_{y_k}}{g(\mathbf{v}_y)} \right) \end{aligned} \tag{53}$$

For simplicity, let's define $d_1^2 = d_{\mathcal{S}^K}^2(\mathbf{x}, \mathbf{y})$ and $d_2^2 = d_{\mathcal{S}^K}^2(\mathbf{x} \oplus \mathbf{v}_x, \mathbf{y} \oplus \mathbf{v}_y)$, then

$$\begin{aligned} d_2^2 &= d_1^2 + d_{\mathcal{S}^K}^2(\mathbf{v}_x, \mathbf{v}_y) + 2 \sum_{k=1}^K \left(\log \frac{x_k}{g(\mathbf{x})} - \log \frac{y_k}{g(\mathbf{y})} \right) \left(\log \frac{v_{x_k}}{g(\mathbf{v}_x)} - \log \frac{v_{y_k}}{g(\mathbf{v}_y)} \right) \\ &= d_1^2 + d_{\mathcal{S}^K}^2(\mathbf{v}_x, \mathbf{v}_y) + 2 \sum_{k=1}^K \log \frac{x_k}{g(\mathbf{x})} \left(\log \frac{v_{x_k}}{v_{y_k}} - \log \frac{g(\mathbf{v}_x)}{g(\mathbf{v}_y)} \right) - 2 \sum_{k=1}^K \log \frac{y_k}{g(\mathbf{y})} \left(\log \frac{v_{x_k}}{v_{y_k}} - \log \frac{g(\mathbf{v}_x)}{g(\mathbf{v}_y)} \right) \end{aligned} \quad (54)$$

Let define $\log \frac{g(\mathbf{v}_x)}{g(\mathbf{v}_y)} = \log \frac{\left(\prod_k v_{x_k} \right)^{1/K}}{\left(\prod_k v_{y_k} \right)^{1/K}} = \frac{1}{K} \sum_k \log \frac{v_{x_k}}{v_{y_k}} = \frac{\Delta}{K}$. Then,

$$d_2^2 = d_1^2 + d_{\mathcal{S}^K}^2(\mathbf{v}_x, \mathbf{v}_y) + 2 \sum_{k=1}^K \log \frac{v_{x_k}}{v_{y_k}} \left(\log \frac{x_k}{g(\mathbf{x})} - \log \frac{y_k}{g(\mathbf{y})} \right) - \frac{2\Delta}{K} \sum_{k=1}^K \left(\log \frac{x_k}{g(\mathbf{x})} - \log \frac{y_k}{g(\mathbf{y})} \right) \quad (55)$$

Since CLR is a zero-mean transformation, $\sum_k \log \frac{x_k}{g(\mathbf{x})} = 0$ and $\sum_k \log \frac{y_k}{g(\mathbf{y})} = 0$. Therefore,

$$d_2^2 = d_1^2 + d_{\mathcal{S}^K}^2(\mathbf{v}_x, \mathbf{v}_y) + 2 \sum_{k=1}^K \log \frac{v_{x_k}}{v_{y_k}} \left(\log \frac{x_k}{g(\mathbf{x})} - \log \frac{y_k}{g(\mathbf{y})} \right) \quad (56)$$

Additionally, $d_{\mathcal{S}^K}^2(\mathbf{v}_x, \mathbf{v}_y) \geq 0$ can be expressed as,

$$\begin{aligned} d_{\mathcal{S}^K}^2(\mathbf{v}_x, \mathbf{v}_y) &= \sum_{k=1}^K \left(\log \frac{v_{x_k}}{v_{y_k}} - \log \frac{g(\mathbf{v}_x)}{g(\mathbf{v}_y)} \right)^2 \\ &= \sum_{k=1}^K \left(\log \frac{v_{x_k}}{v_{y_k}} \right)^2 + \sum_{k=1}^K \left(\log \frac{g(\mathbf{v}_x)}{g(\mathbf{v}_y)} \right)^2 - 2 \log \frac{g(\mathbf{v}_x)}{g(\mathbf{v}_y)} \sum_{k=1}^K \log \frac{v_{x_k}}{v_{y_k}} \\ &= \sum_{k=1}^K \left(\log \frac{v_{x_k}}{v_{y_k}} \right)^2 - \frac{\Delta^2}{K} \end{aligned} \quad (57)$$

Therefore,

$$d_2^2 = d_1^2 + \sum_{k=1}^K \left(\log \frac{v_{x_k}}{v_{y_k}} \right)^2 - \frac{\Delta^2}{K} + 2 \sum_{k=1}^K \log \frac{v_{x_k}}{v_{y_k}} \left(\log \frac{x_k}{g(\mathbf{x})} - \log \frac{y_k}{g(\mathbf{y})} \right) \quad (58)$$

Now, consider $\tau_u = \max\{\log \frac{v_{x_k}}{v_{y_k}}\}$ and $\tau_l = \max\{\log \frac{v_{y_k}}{v_{x_k}}\} = -\min\{\log \frac{v_{x_k}}{v_{y_k}}\}$. Then,

$$\begin{aligned} d_2^2 &\leq d_1^2 + K\tau_u^2 - \frac{\Delta^2}{K} + 2\tau_u \left(\sum_k \log \frac{x_k}{g(\mathbf{x})} - \sum_k \log \frac{y_k}{g(\mathbf{y})} \right) \\ d_2^2 &\geq d_1^2 + K\tau_l^2 - \frac{\Delta^2}{K} - 2\tau_l \left(\sum_k \log \frac{x_k}{g(\mathbf{x})} - \sum_k \log \frac{y_k}{g(\mathbf{y})} \right) \end{aligned} \quad (59)$$

Again, because of $\sum_k \log \frac{x_k}{g(\mathbf{x})} = 0$ and $\sum_k \log \frac{y_k}{g(\mathbf{y})} = 0$, we can conclude that,

$$d_1^2 - \frac{\Delta^2}{K} + K\tau_l^2 \leq d_2^2 \leq d_1^2 - \frac{\Delta^2}{K} + K\tau_u^2 \quad (60)$$

$$d_1^2 - \Gamma_l \leq d_2^2 \leq d_1^2 + \Gamma_u \quad (61)$$

Since $K\tau_u \geq \Delta \geq K\tau_l$, we can conclude $\Gamma_u, \Gamma_l \geq 0$. \square

Proposition 5. *Given samples $\mathbf{x}, \mathbf{y} \in \mathcal{S}^K$, where \mathcal{S}^K is a simplex of K parts, we have*

$$0 \leq d_{\mathbf{v}}^2(\mathbf{x}, \mathbf{y}) - d_{\mathcal{S}^K}^2(\mathbf{x} \oplus \mathbf{v}_x, \mathbf{y} \oplus \mathbf{v}_y) \leq \frac{1}{K}(\Delta + K\tau)^2$$

where $d_{\mathbf{v}}^2(\mathbf{x}, \mathbf{y}) = \sum_k (\log x_k v_{x_k} - \log y_k v_{y_k})^2$, $\tau = \max_k \{\log \frac{x_k}{y_k}\}$, and $\Delta = \sum_k \left(\log \frac{v_{x_k}}{v_{y_k}} \right)$.

Proof.

$$\begin{aligned} d_{\mathcal{S}^K}^2(\mathbf{x} \oplus \mathbf{v}_x, \mathbf{y} \oplus \mathbf{v}_y) &= \sum_{k=1}^K \left(\log x_k v_{x_k} - \log y_k v_{y_k} - \frac{1}{K} \log \prod_k \frac{x_k v_{x_k}}{y_k v_{y_k}} \right)^2 \\ &= \sum_{k=1}^K \left(\log x_k v_{x_k} - \log y_k v_{y_k} - \frac{1}{K} \sum_k \log \frac{x_k v_{x_k}}{y_k v_{y_k}} \right)^2 \\ &= \sum_{k=1}^K (\log x_k v_{x_k} - \log y_k v_{y_k} - D)^2 \end{aligned} \quad (62)$$

where $D = \frac{1}{K} \sum_k (\log x_k v_{x_k} - \log y_k v_{y_k})$. Therefore,

$$\begin{aligned} d_{\mathcal{S}^K}^2(\mathbf{x} \oplus \mathbf{v}_x, \mathbf{y} \oplus \mathbf{v}_y) &= \sum_{k=1}^K (\log x_k v_{x_k} - \log y_k v_{y_k})^2 + KD^2 - 2D \sum_{k=1}^K (\log x_k v_{x_k} - \log y_k v_{y_k}) \\ &= d_{\mathbf{v}}^2(\mathbf{x}, \mathbf{y}) - KD^2 \end{aligned}$$

$$d_{\mathbf{v}}^2(\mathbf{x}, \mathbf{y}) = d_{\mathcal{S}^K}^2(\mathbf{x} \oplus \mathbf{v}_x, \mathbf{y} \oplus \mathbf{v}_y) + KD^2 \quad (63)$$

Since $KD^2 \geq 0$, $d_{\mathbf{v}}^2(\mathbf{x}, \mathbf{y}) \geq d_{\mathcal{S}^K}^2(\mathbf{x} \oplus \mathbf{v}_x, \mathbf{y} \oplus \mathbf{v}_y)$.

Now, considering $\tau = \max_k \{\log \frac{x_k}{y_k}\}$, and $\Delta = \sum_k \left(\log \frac{v_{x_k}}{v_{y_k}} \right)$, then

$$\begin{aligned} d_{\mathbf{v}}^2(\mathbf{x}, \mathbf{y}) - d_{\mathcal{S}^K}^2(\mathbf{x} \oplus \mathbf{v}_x, \mathbf{y} \oplus \mathbf{v}_y) &= \frac{1}{K} \left(\sum_k \left(\log \frac{x_k}{y_k} + \log \frac{v_{x_k}}{v_{y_k}} \right) \right)^2 \\ &= \frac{1}{K} \left(\Delta + \sum_k \left(\log \frac{x_k}{y_k} \right) \right)^2 \\ &\leq \frac{1}{K} (\Delta + K\tau)^2. \end{aligned} \quad (64)$$

\square

Proposition 3. Suppose $\mathbf{c}_a, \mathbf{c}_b \in \mathcal{S}^K$, where \mathcal{S}^K is a simplex of $K > 0$ parts. If $d_{\mathcal{S}^K}(\mathbf{c}_a, \mathbf{c}_b)$ denotes the distance in Aitchison geometry and $d_\sigma^2(\mathbf{c}_a, \mathbf{c}_b) = \sum_k \left(\sigma_{a_k}^{-1} \log c_{a_k} - \sigma_{b_k}^{-1} \log c_{b_k} \right)^2$ denotes a perturbed distance, then

$$d_{\mathcal{S}^K}^2(\mathbf{c}_a, \mathbf{c}_b) - \rho_l \leq d_\sigma^2(\mathbf{c}_a, \mathbf{c}_b) \leq d_{\mathcal{S}^K}^2(\mathbf{c}_a, \mathbf{c}_b) + \rho_u$$

where $\rho_u, \rho_l \geq 0$, $\rho_u = K(\tau_{\sigma_u}^2 + \tau_{\mathbf{c}}^2) + 2\Delta_\sigma \tau_{\mathbf{c}}$, $\rho_l = \frac{\Delta_\sigma^2}{K} - K\tau_{\sigma_l}^2$, $\tau_{\mathbf{c}} = \max_k \{\log c_{a_k} - \log c_{b_k}\}$, $\tau_{\sigma_u} = \max_k \{g_k\}$, $\tau_{\sigma_l} = \max_k \{-g_k\}$, $\Delta_\sigma = \sum_k g_k$, and $g_k = (\sigma_{a_k}^{-1} - 1) \log c_{a_k} - (\sigma_{b_k}^{-1} - 1) \log c_{b_k}$.

Proof. In Propositions 4 and 5, by considering $\mathbf{x} = \mathbf{c}_a$, $\mathbf{y} = \mathbf{c}_b$, $\mathbf{v}_x = \mathbf{v}_a = \left(\frac{(\sigma_{a_1}^{-1}-1)}{\gamma_a}, \dots, \frac{(\sigma_{a_K}^{-1}-1)}{\gamma_a} \right)$, and

$$\mathbf{v}_y = \mathbf{v}_b = \left(\frac{(\sigma_{b_1}^{-1}-1)}{\gamma_b}, \dots, \frac{(\sigma_{b_K}^{-1}-1)}{\gamma_b} \right), \text{ where } \gamma_a = \sum_k c_{a_k}^{(\sigma_{a_k}^{-1}-1)} \text{ and } \gamma_b = \sum_k c_{b_k}^{(\sigma_{b_k}^{-1}-1)}, \text{ we have}$$

$$d_{\mathcal{S}^K}^2(\mathbf{c}_a \oplus \mathbf{v}_a, \mathbf{c}_b \oplus \mathbf{v}_b) = \sum_{k=1}^K \left(\sigma_{a_k}^{-1} \log c_{a_k} - \sigma_{b_k}^{-1} \log c_{b_k} - D \right)^2 \quad (65)$$

where $D = \frac{1}{K} \sum_k \left(\sigma_{a_k}^{-1} \log c_{a_k} - \sigma_{b_k}^{-1} \log c_{b_k} \right)$. Hence,

$$d_{\mathcal{S}^K}^2(\mathbf{c}_a, \mathbf{c}_b) + K\tau_{\sigma_l}^2 - \frac{\Delta_\sigma^2}{K} \leq d_{\mathcal{S}^K}^2(\mathbf{c}_a \oplus \mathbf{v}_a, \mathbf{c}_b \oplus \mathbf{v}_b) \leq d_{\mathcal{S}^K}^2(\mathbf{c}_a, \mathbf{c}_b) + K\tau_{\sigma_u}^2 - \frac{\Delta_\sigma^2}{K} \quad (66)$$

and

$$0 \leq d_\sigma^2(\mathbf{c}_a, \mathbf{c}_b) - d_{\mathcal{S}^K}^2(\mathbf{c}_a \oplus \mathbf{v}_a, \mathbf{c}_b \oplus \mathbf{v}_b) \leq \frac{1}{K} (K\tau_{\mathbf{c}} + \Delta_\sigma)^2 \quad (67)$$

Therefore,

$$d_{\mathcal{S}^K}^2(\mathbf{c}_a, \mathbf{c}_b) + K\tau_{\sigma_l}^2 - \frac{\Delta_\sigma^2}{K} \leq d_\sigma^2(\mathbf{c}_a, \mathbf{c}_b) \leq d_{\mathcal{S}^K}^2(\mathbf{c}_a, \mathbf{c}_b) + \frac{1}{K} \left((K\tau_{\mathbf{c}} + \Delta_\sigma)^2 + K^2\tau_{\sigma_u}^2 - \Delta_\sigma^2 \right) \quad (68)$$

$$d_{\mathcal{S}^K}^2(\mathbf{c}_a, \mathbf{c}_b) - \rho_l \leq d_\sigma^2(\mathbf{c}_a, \mathbf{c}_b) \leq d_{\mathcal{S}^K}^2(\mathbf{c}_a, \mathbf{c}_b) + \rho_u. \quad (69)$$

□

F Type-preserving data augmentation

Augmentation can be considered as a generative process. We seek a generative model that not only learns the data distribution, but also transformations that represent within-class variations in an unsupervised manner. Learning such transformations is generally not straightforward, and requires prior knowledge about the underlying invariances. While conventional transformations such as rotation, scaling, or translation can serve as type-preserving augmentations for many image datasets, they may not capture the richness of the underlying process. Moreover, such augmentation strategies cannot be used when within-class invariance are unknown. Suggested alternatives to conventional augmentations either rely on class label, or are specific to image data^{1 2}.

¹Søren Hauberg, Oren Freifeld, Anders Boesen Lindbo Larsen, John Fisher, and Lars Hansen. Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation. In Artificial Intelligence and Statistics, pp. 342–350, 2016.

²Ayush Jaiswal, Rex Yue Wu, Wael Abd-Almageed, and Prem Natarajan. Unsupervised adversarial invariance. In Advances in Neural Information Processing Systems, pp. 5092–5102, 2018.

To this end, inspired by DAGAN³, we propose an unsupervised type-preserving augmentation using a VAE-GAN-like architecture⁴. We seek a network \mathcal{G} such that a noisy copy, \mathbf{x}_a can be obtained as a variation of the given sample, \mathbf{x} , based on its low dimensional representation that is concatenated with Gaussian noise \mathbf{n} . To prevent the network from disregarding the noise, we formulate the training procedure as the following minmax optimization which uses a discriminator network \mathcal{D} as a regularizer.

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{V}(\mathcal{D}, \mathcal{G}) - \mathcal{R}(\mathcal{G}) + \mathcal{T}_{\alpha}(\mathcal{G}) + \gamma d(\mathcal{G}) \quad (70)$$

where,

$$\mathcal{V}(\mathcal{D}, \mathcal{G}) = \mathbb{E}_{\mathbf{x}} [\log \mathcal{D}(\mathbf{x})] + \mathbb{E}_{\mathbf{x}} [\log (1 - \mathcal{D}(\mathbf{x}_{\mathbf{n}}))] + \mathbb{E}_{\mathbf{x}, \mathbf{n}} [\log (1 - \mathcal{D}(\mathbf{x}_{\mathbf{n}}))] \quad (71)$$

$$\mathcal{R}(\mathcal{G}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} [\log p(\mathbf{x}|\mathbf{z})] \quad (72)$$

$$\mathcal{T}_{\alpha}(\mathcal{G}) = \max (\|\mathbf{x} - \mathbf{x}_{\mathbf{n}}\|_2 - \|\mathbf{x} - \mathbf{x}_{\mathbf{a}}\|_2 + \alpha, 0) \quad (73)$$

$$d(\mathcal{G}) = D_{KL}(q(\mathbf{z}|\mathbf{x}) \| q(\mathbf{z}|\mathbf{x}, \mathbf{n})) . \quad (74)$$

While training, \mathcal{G} generates two samples: $\mathbf{x}_{\mathbf{n}}$ and $\mathbf{x}_{\mathbf{a}}$. The former denotes \mathbf{x}_a , and the latter is a sample generated in the absence of noise. In Eq. 70, \mathcal{V} is the value function for the joint training of the discriminator and generator; \mathcal{R} is the reconstruction loss, which operates only over $\hat{\mathbf{x}}$; $\mathcal{T}_{\alpha}(\mathcal{G})$ is the triplet loss that prevents network \mathcal{G} from disregarding noise and generating identical samples; and $d(\mathcal{G})$ is the distance between the latent variables in the absence and presence of noise. $d(\mathcal{G})$ is a regularizer to encourage original and noisy samples to be located close to one another in the latent space and is controlled by hyperparameter $\gamma \ll 1$.

It should be noted that, here the augmenter learns to generate samples in the vicinity of a given sample in the latent space, which is independent of the \mathbf{s} and \mathbf{c} . The augmenter does not use any label information in any way. It is called type-preserving augmenter not because label information was utilized during training or label preservation is guaranteed, but because ‘similar’ samples often belong to the same cluster.

Fig. S1 illustrates the network design for the type-preserving data augmentation for image datasets. For scRNA-seq datasets, we used the similar design that is used for a single arm in cpl-mixVAE (Fig. S13b), without mixture representation, only a continuous space, with $|\mathbf{z}| = 10$.

F.1 Data augmentation for an image dataset: MNIST

Fig. S2 displays example noisy samples generated by the type-preserving augmentation for MNIST. To quantitatively evaluate the proposed type-preserving data augmentation, we used a benchmark classifier for MNIST digits, which achieves 99.54% accuracy over 10,000 test samples⁵. Applying the imported classifier to the augmented test samples yields 96.14% classification accuracy, which demonstrates that the augmenter preserves the label information (type) for **96.58%** of the augmented samples.

F.2 Data augmentation for a non-image dataset: scRNA-seq

Generating augmented samples with the same class identity in the absence of within-class invariance is fairly challenging. In case of image datasets, e.g. MNIST, since there exist some intuitions about the identities of discrete and continuous variational factors, we can explicitly define a set of transformation such as rotation, translation, scaling, flipping, etc. that can be used as type-preserving augmentation. However, for non-image datasets, e.g. the single cell RNA-seq dataset, suggested alternative methods may fail to represent the class-conditioned variation in an unsupervised manner. Moreover, in case of biological datasets, learning an augmentation transformation is rather challenging due to the limited number of samples. Accordingly, in this section, we study the performance of the proposed data augmentation to investigate the extent to which our method is successful in realistic generation of the single-cell RNA-seq samples.

³Antoniou, Antreas, Amos Storkey, and Harrison Edwards. "Data augmentation generative adversarial networks." arXiv preprint arXiv:1711.04340, 2017.

⁴Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In International conference on machine learning, pp. 1558–1566. PMLR, 2016.

⁵Digit Recognizer, kaggle competition: <https://www.kaggle.com/c/digit-recognizer>

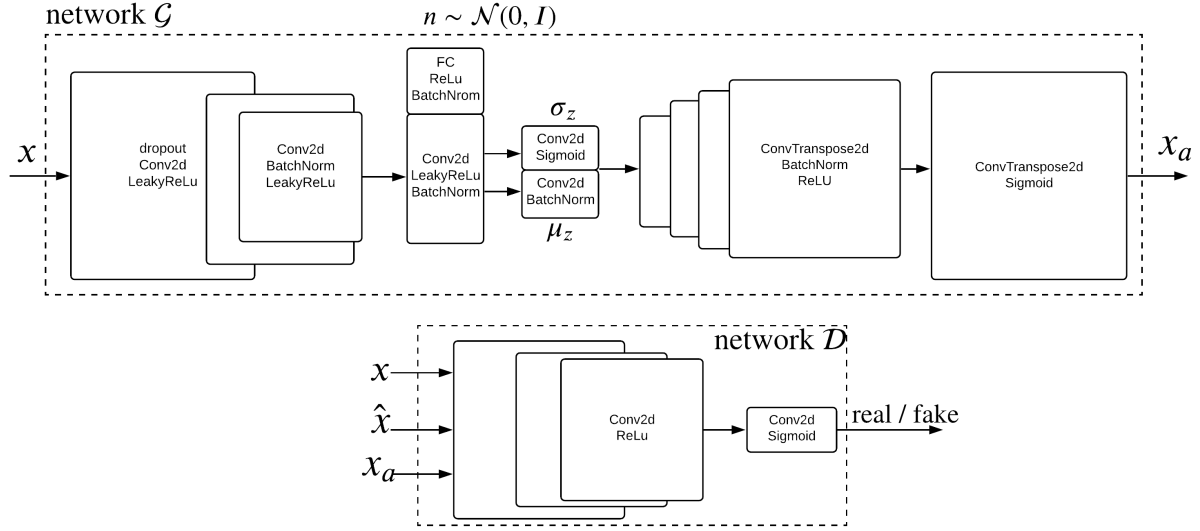


Figure S1: Network architecture for the proposed type-preserving data augmentation for image datasets.

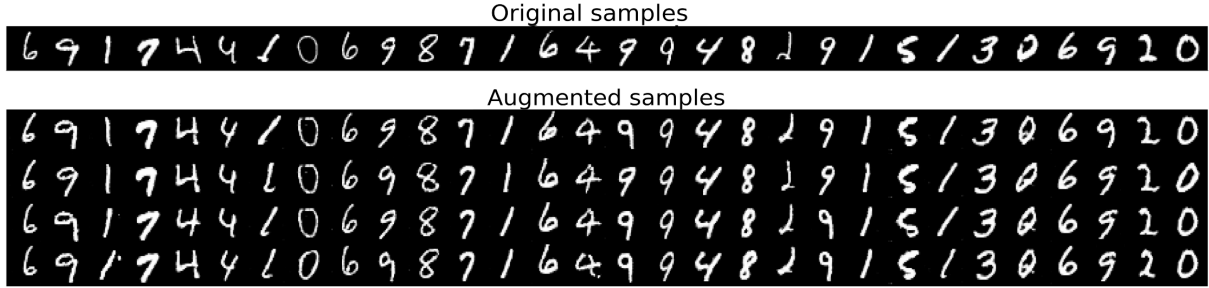


Figure S2: Augmented samples for the MNIST dataset generated by the type-preserving augmentation conserve type of the original sample.

Fig. S3 illustrates a two-dimensional demonstrations for both original and augmented single cells samples. For two-dimensional visualizations, here, we used a regular autoencoder for non-linear dimension reduction. First, the autoencoder has been trained on the original cell samples. After learning a two-dimensional coordinate system for the original samples (left panel), we used the autoencoder to visualize the augmented samples (right panel). Comparing the visualizations demonstrates that the representations are qualitatively similar and all groups of cells sharing the same type (same color) are placed in similar locations. Additionally, in Fig. S4, we show the expression profiles of a subset of genes for an inhibitory cell. Again the qualitative comparison of the expression profiles reveals a similar variability across genes. Since the single cell RNA-seq data is heavily unbalanced, we additionally reported the data augementer’s performance at the single gene expression level. Fig. S5 illustrates the expression distribution of a subset of known genes for augmented cell samples (colorful histograms) compared with the original expressions (gray histograms).

G Sensitivity of representation learning to the hyperparameters

The cpl-mixVAE framework, similar to other deep neural network approaches, has a regularization hyperparameter λ which controls coupling among a pair of autoencoder arms. In this section, we have conducted a series of experiments to assess the sensitivity of the cpl-mixVAE’s performance to its coupling factor, in comparison with JointVAE which has four critical hyperparameters, two for the discrete and two for the

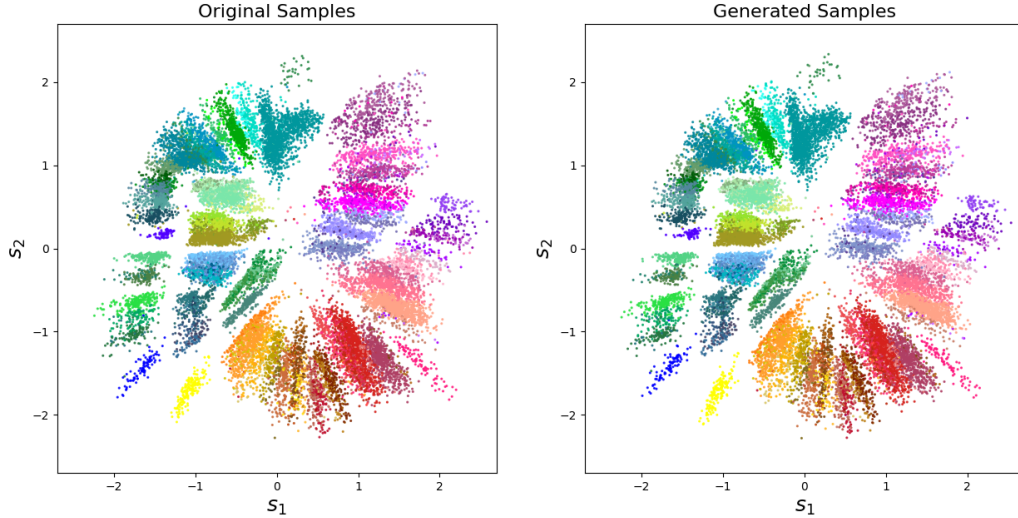


Figure S3: Evaluation of the generated sample by the proposed type-preserving data augmenter. Both figures represent a low dimensional visualization of single cells samples that are obtained from a regular autoencoder that is only used for non-linear dimension reduction. Left panel shows the original Smart-seq ALM-VISp dataset in a two-dimensional space by means of two coordinates. Right panel visualizes the generated cell samples by the augmenter in the same coordinate system. Both visualizations are obtained for 22,000 cell samples with 5,000 genes, and 115 neuron types. The color code is assigned according to the proposed taxonomy in Tasic et al., 2018.

continuous variables. Fig. S6 shows how the mixture representation performance changes for both JointVAE and cpl-mixVAE by changing their hyperparameters. For JointVAE, here, we only consider the channel capacity for the discrete variable, i.e. C_c , which requires adjustment over training iterations.

Fig. S6a shows changes of the categorical assignment accuracy as a function of λ (for cpl-mixVAE) and C_c (for JointVAE). While cpl-mixVAE’s performance is adequate for different values of the coupling factor, JointVAE is susceptible to the changes of the channel capacity factor. Although encoding channel capacity (as an estimation for mutual information) for each dataset with different latent space dimension and training iterations is computationally expensive, a main problem of using these hyperparameters happens when the learning of the model is highly sensitive to the channel capacity. For instance, Fig. S6b illustrates the categorical variables learned by JointVAE, when we reduced the maximum capacity from 5 to 1. Likewise, Fig. S6d shows a similar learning issue for JointVAE, when increasing the maximum channel from 5 to 25. In case of cpl-mixVAE, we can see that although obtaining the best performance requires parameter tuning, the model acceptably works with any empirical choice of $\lambda \in [0.1, 10]$.

H MNIST dataset analysis

A common assumption in “disentangling” the continuous and discrete factors of variability is the independence of the categorical and continuous latent variables, conditioned on data. Fig. S7 demonstrates that this assumption can be significantly violated for two commonly used, interpretable style variables, “angle” and “width,” in the MNIST dataset.

Calculation of angle and width: We first calculate the inertia matrix for each sample by treating the image as a solid object with a mass distribution given by pixel brightness values. Then, we compute the principal axis of the image based on the inertia matrix. We report the angle between this vector and the vertical axis using the $[-\pi/2, \pi/2)$ range. To calculate the width, we project the image to the horizontal axis after aligning the principal axis with the vertical axis using the computed angle value. We report the support of this projected signal, normalized by the horizontal size of the image (here 28 pixels).

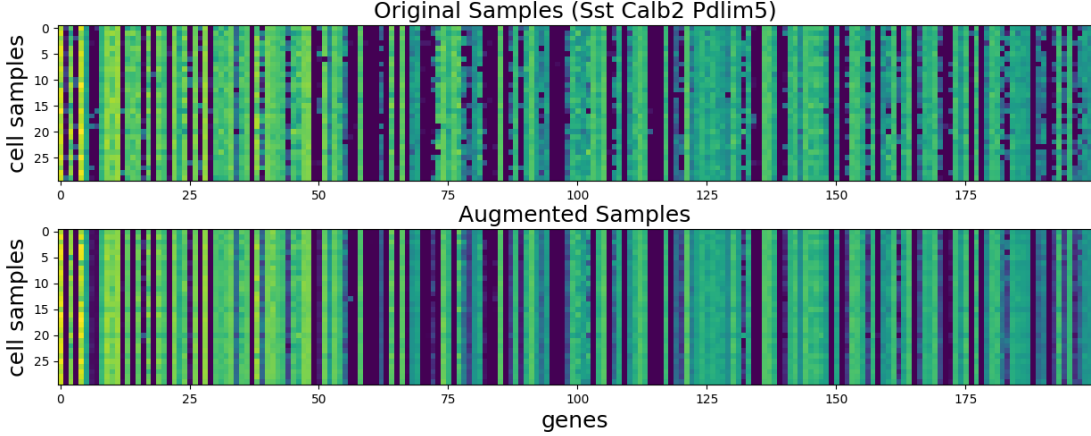


Figure S4: Qualitative comparison across the original and augmented gene expression profiles for an inhibitory Sst type cell.

I Dependence of state and class label in JointVAE

We analyzed the effects of the dependency between the continuous and discrete latent factors on the results obtained by state-of-the-art methods for joint representation learning, e.g. JointVAE or CascadeVAE. These methods formulate the continuous and discrete variability as two independent factors such that the discrete factor is expected to determine the cluster to which each sample belongs, while the continuous factor represents the *class-independent* variability. In many applications, however, the assumption of a discrete-continuous dichotomy may not be satisfied. (Section H analyzes this assumption for the MNIST dataset.)

Fig. S8a illustrates four dimensions of the continuous latent variable \mathbf{s} obtained by the JointVAE model for the MNIST dataset. Here, colors represent the digit type of each \mathbf{s} sample. While the prior distribution is assumed to be Gaussian, the dependency of $\mathbf{s}|\mathbf{x}$ on the digit type, \mathbf{c} , is visible. To quantify this observation, we applied an unsupervised clustering method, i.e. Gaussian mixture model (GMM) with 10 clusters, to the continuous RV samples obtained from a JointVAE network trained for 150000 iterations. This unsupervised model achieved an overall clustering accuracy of **66%**. Fig. S8b shows the results for individual digits, e.g. 83% for digit “1” (Fig. S8). Together, these results demonstrate the violation of the independence assumption for $q(\mathbf{s}|\mathbf{x})$ and $q(\mathbf{c}|\mathbf{x})$.

J Ablation studies

J.1 MNIST

As discussed earlier in Section 4.3, to show how the A -arm VAE framework is successful in mixture modeling, we investigated the categorical assignment performance under different training settings. Since CascadeVAE does not learn the categorical factors by variational inference, here we only studied JointVAE as a 1-arm VAE, and cpl-mixVAE as a 2-arm VAE. Table S1 shows the performance of JointVAE, cpl-mixVAE and their variants under different training settings for the MNIST dataset. Here, we considered the reconstruction error, i.e. \mathcal{L}_{rec} and accuracy of the categorical performance (ACC). In Table S1, JointVAE denotes the average performance for the original JointVAE that has been trained by settings suggested in (Dupont, 2018); JointVAE[†], is the JointVAE model that has been trained with noisy copies of the original MNIST dataset generated by the type-preserving data augmentation method in Section F; JointVAE[‡] is another JointVAE model that uses the same architecture for the basic encoder/decoder networks as the one used in cpl-mixVAE. Our results does not show any improvement in the performance of JointVAE by using data augmentation or altering the network architecture.

Next, we studied the performance changes of the proposed 2-arm cpl-mixVAE under three different settings. In Table S1, cpl-mixVAE is the proposed 2-arm VAE framework using coupled-autoencoders and the type-

| Method | $\mathcal{L}_{\text{rec}} \downarrow$ | ACC \uparrow (mean \pm s.d.) |
|--|---------------------------------------|-------------------------------------|
| JointVAE | 0.166 | 68.99 \pm 11.76 |
| JointVAE [†] | 0.166 | 68.21 \pm 09.58 |
| JointVAE [‡] | 0.162 | 62.19 \pm 05.73 |
| cpl-mixVAE | 0.145 | 84.56 \pm 06.47 |
| cpl-mixVAE* | 0.140 | 80.25 \pm 05.37 |
| cpl-mixVAE ^a | 0.135 | 82.92 \pm 04.64 |
| cpl-mixVAE(s \nparallel c) | 0.146 | 79.63 \pm 08.32 |

Table S1: Categorical assignment in the mixture representation learning under different training settings for 10 randomly initialized runs, for the MNIST dataset.

preserving data augmentation in Section F; cpl-mixVAE*, is a cpl-mixVAE in which coupled networks are not independent and the networks parameters are shared; cpl-mixVAE^a, is a cpl-mixVAE model that uses random rotations ($[-\pi/9, \pi/9]$) as an for data augmentation; and lastly cpl-mixVAE(**s** \nparallel **c**), is a cpl-mixVAE model in which the state variable is independent of the discrete variable. Our results show that the proposed cpl-mixVAE obtained the best categorical assignment among all training settings.

Furthermore, we investigate the performance of cpl-mixVAE for different cardinalities of the categorical variable, **c**. Fig. S9a shows the performance of cpl-mixVAE in terms of the log-ratio of $\mathbb{E}_{\mathbf{x}} [\max_{\mathbf{c}} q(\mathbf{c}|\mathbf{x})] / \mathbb{E}_{\mathbf{x}} [\min_{\mathbf{c}} q(\mathbf{c}|\mathbf{x})]$ as a function of $|\mathbf{c}| \in [7, 15]$. Our results demonstrate that an insufficient number of categories results in inaccurate encoding of discrete variability, leading to some dimensions being allocated to more than one digit (Fig. S9b.1), which results in a lower log-ratio. On the other hand, additional c_k leaves some categories under-utilized (Fig. S9b.3 and Fig. S9b.4), which again leads to a lower log-ratio value. Notably, our results show that while JointVAE suffers from sensitivity to empirical choices of $|\mathbf{c}|$ (Dupont, 2018), cpl-mixVAE is more robust in encoding the discrete variability, and the log-ratio measure is maximized at $|\mathbf{c}| = 10$. For $|\mathbf{c}| < 10$, cpl-mixVAE utilizes all categories, without suffering from mode collapse and for $|\mathbf{c}| > 10$, it does not allocate unneeded categories and maintains high categorical assignment accuracy.

J.2 scRNA-seq

Here, we examine the accuracy of categorical assignments for VAE-based mixture models, under different dimensions of discrete space. For this purpose, we merged neuron types in the Smart-seq ALM-VISp dataset using hierarchical taxonomy defined by Tasic et al., 2018. Fig. S10a illustrates the cell type taxonomy for the Smart-seq dataset. The dendrogram shows the hierarchical relationship between 115 neuron types, where the first bifurcation from the top represents the split between inhibitory (right branch) and excitatory neurons (left branch). We used the hierarchical dendrogram to assess the performance of the 1-arm and 2-arm VAEs at different levels of cell type taxonomy. First, we obtained a smaller number of cell classes by progressively merging the 115 types according to the hierarchical dendrogram. For instance, at the three merging levels from bottom to top, indicated in Fig. S10a, we obtain 57, 10, and 2 distinct merged neuron sub-classes, respectively.

Fig. S10b compares the performances of JointVAE and cpl-mixVAE(2-arm) for different numbers of categories obtained for merged types. For instance, $|\mathbf{c}| = 2$ corresponds to the highest node in the dendrogram, where there are only two categories. As expected, the accuracy significantly increases as types are merged according to the hierarchy. Consistent with the results for the MNIST dataset, once again cpl-mixVAE outperforms the JointVAE model. Note that here, chance level is estimated based on the most abundant type in the dataset.

Additionally, here, we report the clustering performances of both VAE arms in Fig. S10b. Our results demonstrate that the performances of both arms are very similar, suggesting that they identify similar types with comparable accuracy.

K Additional study for scRNA-seq data

K.1 Robustness of type-dependent latent factors in cpl-mixVAE

To address the consistency of our results for the continuous latent factor, i.e., \mathbf{s} over multiple runs, in Fig. S11, we show the continuous variable traversals for a subset of genes including MGs, IEGs, and HKGs, across randomly initialized runs, for “L5 NP” cell types in the ALM area. Each sub-figure illustrates the normalized reconstructed gene expression with respect to the latent continuous factor. As discussed in Section 4.2, the normalized expression of the MGs i.e., *Slc38a11*, *Slc383a*, and *Foxc1* is unaffected by changes of identified continuous variables. In contrast, for IEGs and HKGs, the changes in normalized expression, in almost all cases, is monotonically linked to \mathbf{s} , confirming that it depends strongly on the cell activity variations under different metabolic and environmental conditions.

K.2 Identifying genes regulating inhibitory cell activity

In addition to the continuous traversal study for excitatory neurons presented in Sec. 4.2 of the manuscript, here we examine the role of the continuous latent variable for a subset of inhibitory neurons, i.e. *somatostatin* neurons known as “Sst” cells. Similar to Fig. 4, for a given cell and its discrete type, we changed the continuous factor using the conditional distribution, and inspected gene expression changes caused by continuous variable alterations. Fig. S12 shows the results of the continuous traversal study for cpl-mixVAE, for 21 Sst cell types belonging to ALM and VISp regions in the brain. In each sub-figure, each column belongs to one Sst type and each panel represents the latent traversal for one gene. The latent traversal is color-mapped to normalized reconstructed expression values, where the y -axis corresponds to the continuous variable. Fig. S12a depicts 7 known marker genes for Sst cells, and Fig. S12b and Fig. S12c correspond to 10 immediate early genes (IEG) and housekeeping genes (HKG) subgroups for inhibitory cells. The normalized expression of the reported MGs as indicators for all Sst types (discrete factors) is unaffected by changes of identified continuous variables. Moreover, Similar to the excitatory cells, the expression changes inferred by cpl-mixVAE for IEGs and HKGs are monotonically linked to the continuous variable, and depend on the cell type.

L Implementation and training settings

L.1 Architecture of the networks

Fig. S13 shows the network architecture for the 2-coupled mixVAE model applied on the benchmark datasets, e.g. MNIST (Fig. S13a), and scRNA-seq datasets (Fig. S13b), respectively. In this architecture, each VAE arm received non-identical copies of the original sample.

For all dataset, To enhance the training process, we also applied random dropout of the input sample and the state variable, respectively.

For MNIST and dSprites, JointVAE and CascadeVAE have been trained by using the same network design and training parameters suggested in (Dupont, 2018; Jeong & Song, 2019).

For InfoGAN, we also used the same network design and parameter setting suggested in (Chen et al., 2016) for the MNIST dataset.

JointVAE[†] uses the same network architecture as a single arm of cpl-mixVAE. That is, it still uses the same loss function and learning procedure as JointVAE, but its convolutional layers are replaced by fully-connected layers, to demonstrate that these architecture choices do not explain the improvement achieved by cpl-mixVAE.

Following, we provide the details of the training parameters for introduced models, for each dataset. Note that to calculate the computational cost of each method (Table 1), we unified the batch size and size of data (e.g., image size) across all methods, and reported the execution time of each iteration on the same GPU machine.

L.2 Training parameters for the dSprites dataset

Training details used for the dSprites dataset are listed as follows.

cpl-mixVAE

- Continuous and categorical variational factors: $|\mathbf{s}| = 6$, $|\mathbf{c}| = 3$
- Batch size: 256
- Training epochs: 300
- τ : 0.67
- λ (coupling weight): 1
- Optimizer: Adam with learning rate $1\text{e-}4$
- Network parameters: 19,602,102
- Training iterations: 734,400

L.3 Training parameters for the MNIST dataset

Training details used for the MNIST dataset are listed as follows. For JointVAE[†] and JointVAE[‡] model, we used the same training parameters as reported in (Dupont, 2018).

cpl-mixVAE

- Continuous and categorical variational factors: $|\mathbf{s}| = 10$, $|\mathbf{c}| = 10$
- Batch size: 256
- Training epochs: 500
- τ (temperature for sampling from Gumbel-softmax distribution): 0.67
- λ (coupling weight): 1
- Optimizer: Adam with learning rate $1\text{e-}4$
- Network parameters: 2,328,204
- Training iterations: 140,400

L.4 Training parameters for scRNA-seq datasets

Training details used for both scRNA-seq datasets are listed as follows. For the JointVAE and CascadeVAE models, we used the same network architecture and data augmentation employed for each arm in cpl-mixVAE (Fig. S13b). The reported training parameters for JointVAE corresponds to the best performance that we obtained for this model.

cpl-mixVAE (Smart-seq ALM-VISp)

- Continuous and categorical variational factors: $|\mathbf{s}| = 2$, $|\mathbf{c}| = 115$
- Batch size: 1000
- D (size of the last hidden layer): 10
- Training epochs: 10000

- τ : 1
- λ (coupling weight): 1
- Optimizer: Adam with learning rate 1e-3
- Network parameters: 2, 156, 132
- Training iterations: 200,000

cpl-mixVAE (10X MOp)

- Continuous and categorical variational factors: $|\mathbf{s}| = 2$, $|\mathbf{c}| = 140$
- Batch size: 1000
- D (size of the last hidden layer): 10
- Training epochs: 15000
- τ : 1
- λ (coupling weight): 1
- Optimizer: Adam with learning rate 1e-3
- Network parameters: 3, 153, 268
- Training iterations: 1, 545, 000

JointVAE (Smart-seq ALM-VISp)

- Continuous and categorical variational factors: $|\mathbf{s}| = 2$, $|\mathbf{c}| = 115$
- Batch size: 1000
- D (size of the last hidden layer): 10
- Training epochs: 10000
- τ : 1
- γ_s, γ_c (hyperparameters of KL divergence): 100
- $C_s \in \mathbb{R}^2$ (continuous channel capacities): Increased linearly from 0 to 50 in 10000 iterations
- $C_c \in \mathbb{R}^{115}$ (discrete channel capacities): Increased linearly from 0 to 10 in 10000 iterations
- Optimizer: Adam with learning rate 1e-3
- Network parameters: 1, 070, 299
- Training iterations: 200,000

JointVAE (10X MOp)

- Continuous and categorical variational factors: $|\mathbf{s}| = 2$, $|\mathbf{c}| = 140$
- Batch size: 1000
- D (size of the last hidden layer): 10
- Training epochs: 15000
- τ : 1

- $\gamma_{\mathbf{s}}, \gamma_{\mathbf{c}}$ (hyperparameters of KL divergence): 100
- $C_{\mathbf{s}} \in \mathbb{R}^2$ (continuous channel capacities): Increased linearly from 0 to 50 in 20000 iterations
- $C_{\mathbf{c}} \in \mathbb{R}^{140}$ (discrete channel capacities): Increased linearly from 0 to 15 in 20000 iterations
- Optimizer: Adam with learning rate 1e-3
- Network parameters: 1,575,824
- Training iterations: 1,545,000

CascadeVAE (Smart-seq ALM-VISp)

- Continuous and categorical variational factors: $|\mathbf{s}| = 2, |\mathbf{c}| = 115$
- Batch size: 1000
- D (size of the last hidden layer): 10
- Training epochs: 10000
- λ' (hyperparameter of $D_{KL}(q(\mathbf{c}|\mathbf{x}) \parallel U(|\mathbf{c}|))$): 0.1
- β (hyperparameter of $D_{KL}(q(\mathbf{s}|\mathbf{x}) \parallel p(\mathbf{s}))$): Increased linearly from 0 to 10 in 10000 iterations
- Optimizer: Adam with learning rate 1e-4
- Network parameters: 1,069,020
- Training iterations: 200,000

CascadeVAE (10X MOp)

- Continuous and categorical variational factors: $|\mathbf{s}| = 2, |\mathbf{c}| = 140$
- Batch size: 1000
- D (size of the last hidden layer): 10
- Training epochs: 15000
- λ' (hyperparameter of $D_{KL}(q(\mathbf{c}|\mathbf{x}) \parallel U(|\mathbf{c}|))$): 0.1
- β (hyperparameter of $D_{KL}(q(\mathbf{s}|\mathbf{x}) \parallel p(\mathbf{s}))$): Increased linearly from 0 to 10 in 20000 iterations
- Optimizer: Adam with learning rate 1e-4
- Network parameters: 1,080,404
- Training iterations: 1,545,000

M Consensus discrete cell types versus the reference transcriptomic taxonomy

Here, we compare the consensus categorical variables obtained from the proposed multi-arm VAE framework with the reference transcriptomic taxonomy obtained by the hierarchical clustering in Tasic et al., 2018, using the Silhouette score. We first computed the top K principal components (PCs) for the single-cell dataset to reduce the data dimensionality; we then used two sets of cluster labels (cpl-mixVAE categories and hierarchical labels) to compute the average Silhouette scores for each label. Fig. S14 illustrates the average Silhouette scores for the SMART-seq ALM-VISp dataset using cpl-mixVAE and Tasic’s neuronal labels. Higher Silhouette scores (close to 1) suggest that the clusters are better separated. Lower (negative) values suggest clusters are not distinct and separable enough. Note that it is expected for both silhouette scores and the gap between them to decrease when including more PCs (more noise). This figure shows that for different number of PCs, i.e., $K \in \{5, 10, 15, 20\}$, the cpl-mixVAE neuronal categories are better cluster labels than Tasic’s labels, having obtained higher average Silhouette scores for most neuronal types. This analysis suggests that the categorical variables inferred by cpl-mixVAE characterize the discrete classes of neuronal types better than the clusters obtained by the transcriptomic taxonomy.

N Limitations

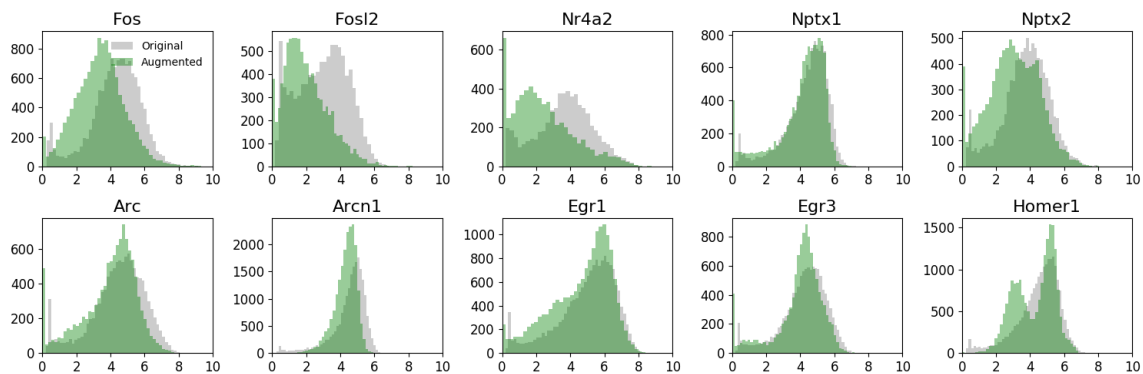
Mixture modeling approaches attempt to infer interpretable discrete and continuous factors of variability that can uncover both type and style (Section 2), wherein lies a main distinction between cpl-mixVAE and object detection or clustering algorithms: clustering algorithms typically specialize in uncovering only the class label.

The introduced generalized mixture model in Fig. 1b assumes that the underlying data manifold can be faithfully represented via a discrete variable that denotes the cluster identity (type), and a continuous variable that denotes the within-cluster variability (style). Consequently, datasets that deviate from this model are ill-suited for cpl-mixVAE. For instance, in the SVHN dataset (recognizing digits in natural scene images), for each category (target digit, here, at the center of the image), there exists multiple discrete and continuous variabilities, such as the presence of other digits next to the target digit (number “526” in Fig. S15). Thus, cpl-mixVAE is not an appropriate choice for SVHN if it is desirable that the inferred categories correspond to digit identities.

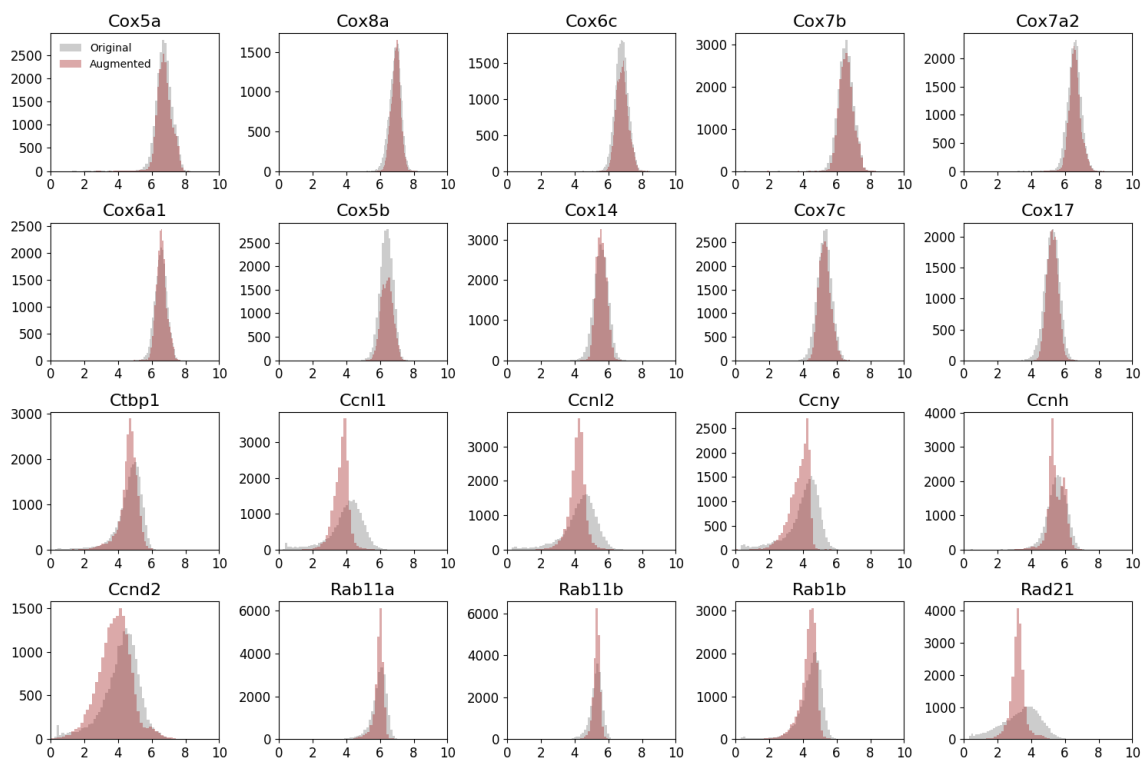
Table S2 displays the performance of the discussed mixture VAE models, JointVAE and cpl-mixVAE for MNIST and SVHN datasets. While cpl-mixVAE performs better than JointVAE and DEC (Xie et al., 2016), which is a clustering method with relatively similar performance for the MNIST dataset, the cpl-mixVAE performance for the SVHN dataset is significantly lower than that for the MNIST dataset. Finally, it may be worth emphasizing that while SVHN is typically conceptualized as “MNIST with street view”, it has a different underlying structure and the within-cluster variability for each digit cannot be accurately modeled only by a continuous variable (Fig. S15).

| Method | Approach | Parameters | MNIST | SVHN |
|----------------------------|------------------|--|-------|-------|
| JointVAE | mixture modeling | $ \mathbf{c} = 10, \mathbf{s} = 10$ | 68.99 | 23.10 |
| cpl-mixVAE | mixture modeling | $ \mathbf{c} = 10, \mathbf{s} = 10$ | 84.56 | 28.89 |
| DEC (Xie et al., 2016) | clustering | $k = 10$ | 84.30 | 11.90 |
| IMSAT (Hu et al., 2017) | clustering | $k = 10$ | 98.40 | 57.30 |
| ACOL (Kilinc et al., 2018) | clustering | $k = 10$ | 98.32 | 76.80 |

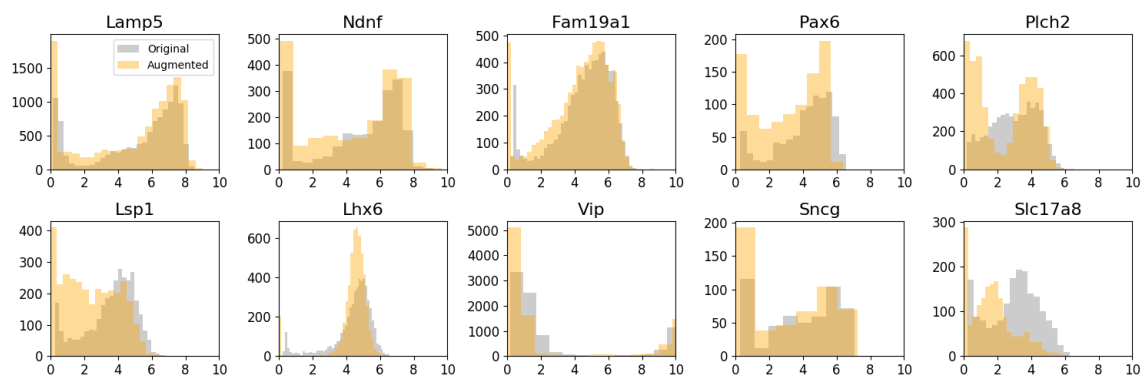
Table S2: Categorical assignment in the mixture representation learning compared to clustering algorithms, for MNIST and SVHN datasets. Here, similar training settings that was used for the MNIST dataset (Section L3), has been used for training both JointVAE and cpl-mixVAE for 286,000 training iterations.



(a) Immediate early genes (IEG)

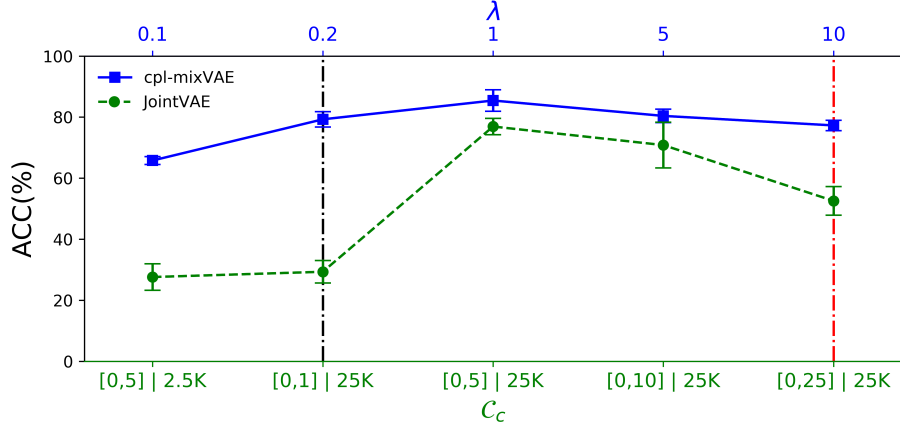


(b) House keeping genes (HKG)



(c) Marker Genes

Figure S5: Comparison between the distribution of genes in the original cell sample (gray color in all figures) and augmented samples for some biologically important subset of genes including (a) immediate early genes (green), (b) house keeping genes (brown), and (c) marker genes (yellow).



(a)

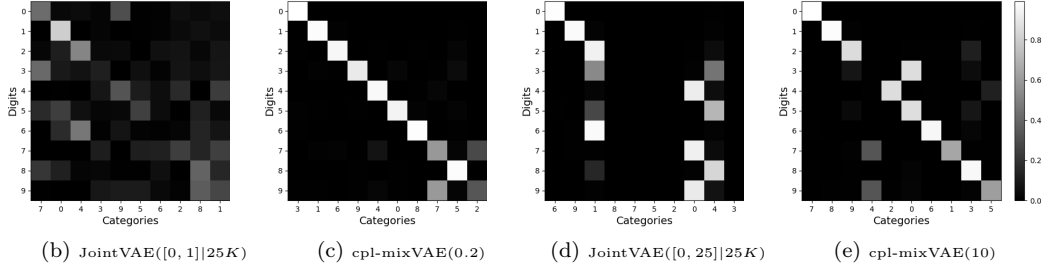


Figure S6: (a) Effect of the coupling factor (λ) in cpl-mixVAE and the channel capacity (C_s) in the JointVAE models. Reported values present the average accuracy of categorical assignment for 3 randomly initialized runs, over 15K training iteration, for the MNIST dataset. (b-c) Confusion matrices for JointVAE and cpl-mixVAE models, respectively corresponding to the hyperparameters marked by the dash-dotted black line. (d-e) Confusion matrices for JointVAE and cpl-mixVAE models, respectively corresponding to the hyperparameters marked by the dash-dotted red line.

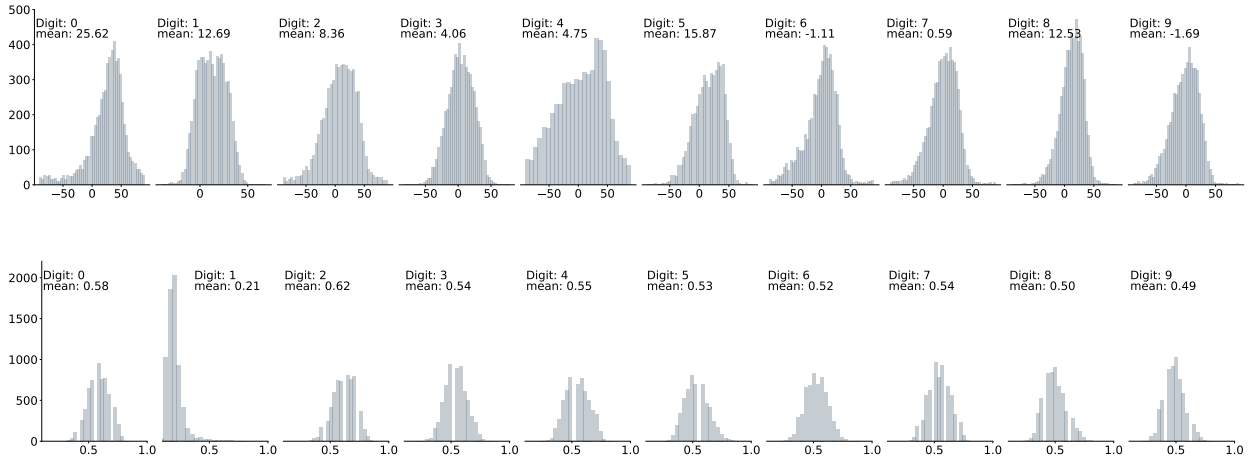


Figure S7: Histograms of angle and width for all digits in MNIST dataset. The empirical distributions of rotation (top) angle and character width (bottom) are illustrated. Comparing the reported mean values and the shape of the histograms demonstrates the dependency of the state variable on the digit type.

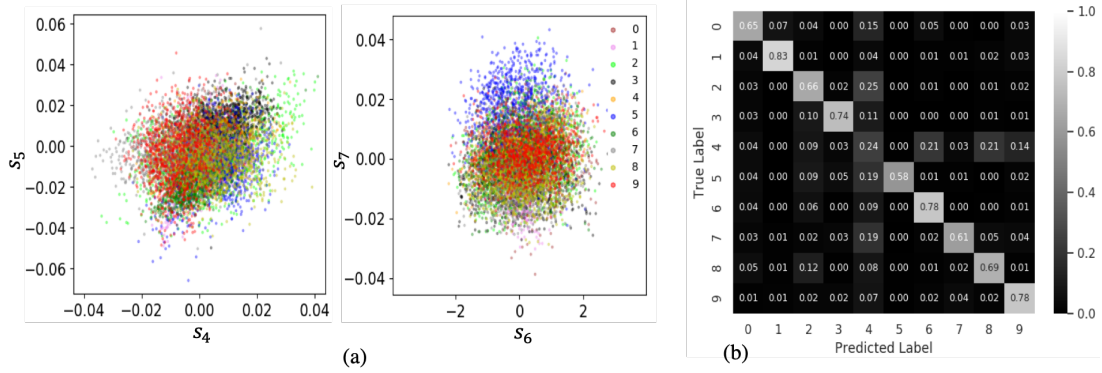


Figure S8: (a) 2-dimensional projections of the continuous variable obtained by JointVAE. Each dot represents a sample of the MNIST dataset and colors represent different digits. (b) Confusion matrix for MNIST digit clustering via GMM using only the continuous latent variable learned by JointVAE.

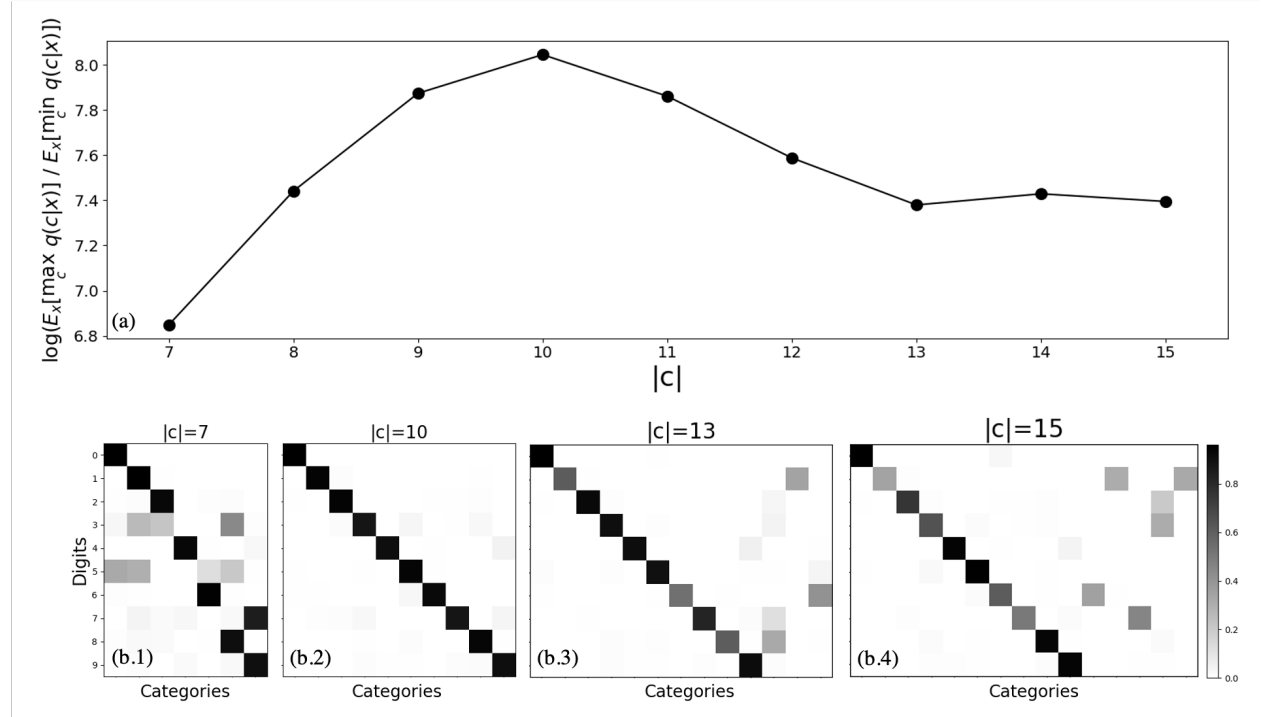
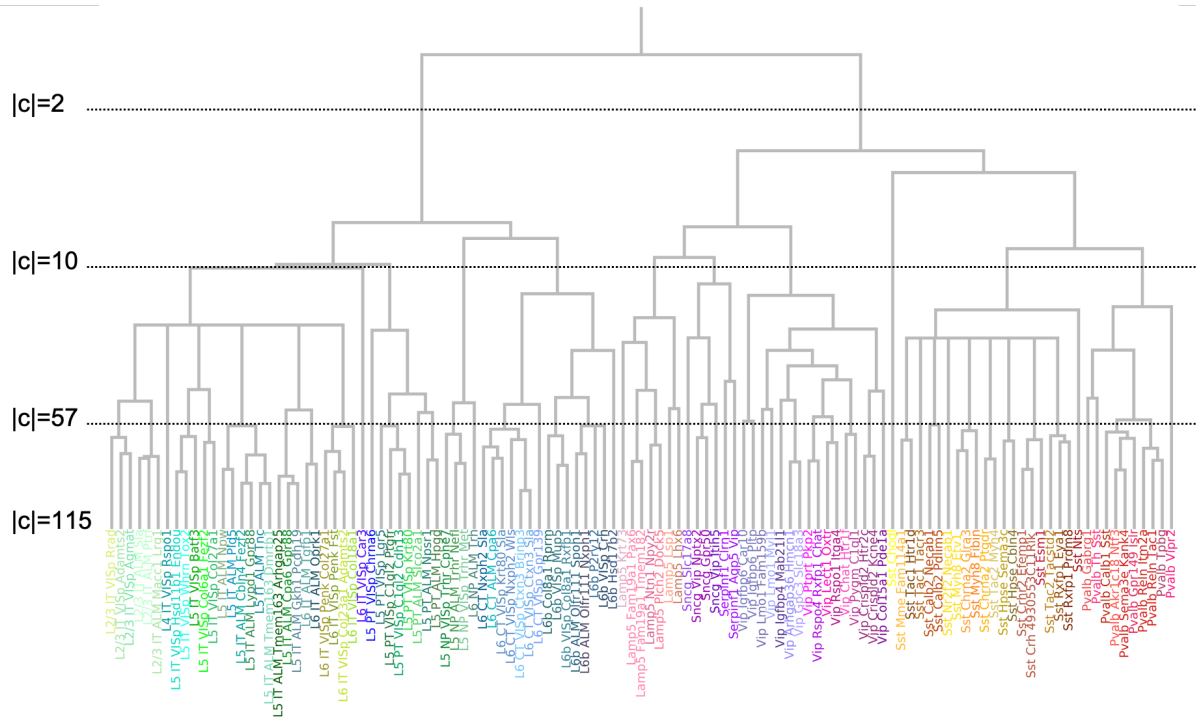
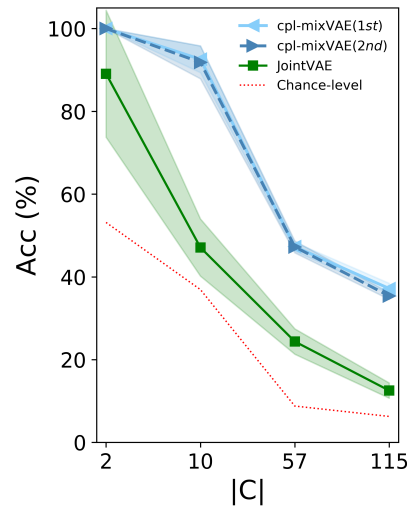


Figure S9: Categorical assignment in the MNIST dataset, when the number of discrete variable ($|c|$) is not equal to the true number of clusters (10 in this case). (a) The plot denotes log-ratio of the average maximum categorical posterior probability to the average minimum posterior probability as a function of $|c|$, for the first arm of cpl-mixVAE. (b1-4) Confusion matrices for the first arm cpl-mixVAE for $|c|$ equal to 7, 10, 13, and 15, respectively.



(a) Taxonomic hierarchy



(b) Categorical assignment

Figure S10: (a) Hierarchical taxonomy of neuron types in the Smart-seq ALM-VISp dataset. Black dotted lines show the merging level of the dendrogram to obtain pre-defined number of discrete types (57, 10, and 2) from the hierarchical taxonomy. (b) Accuracy of categorical assignment for both arms in the cpl-mixVAE and JointVAE as function of number of categories based on the Smart-seq dataset. Red dotted line shows chance-level accuracy.

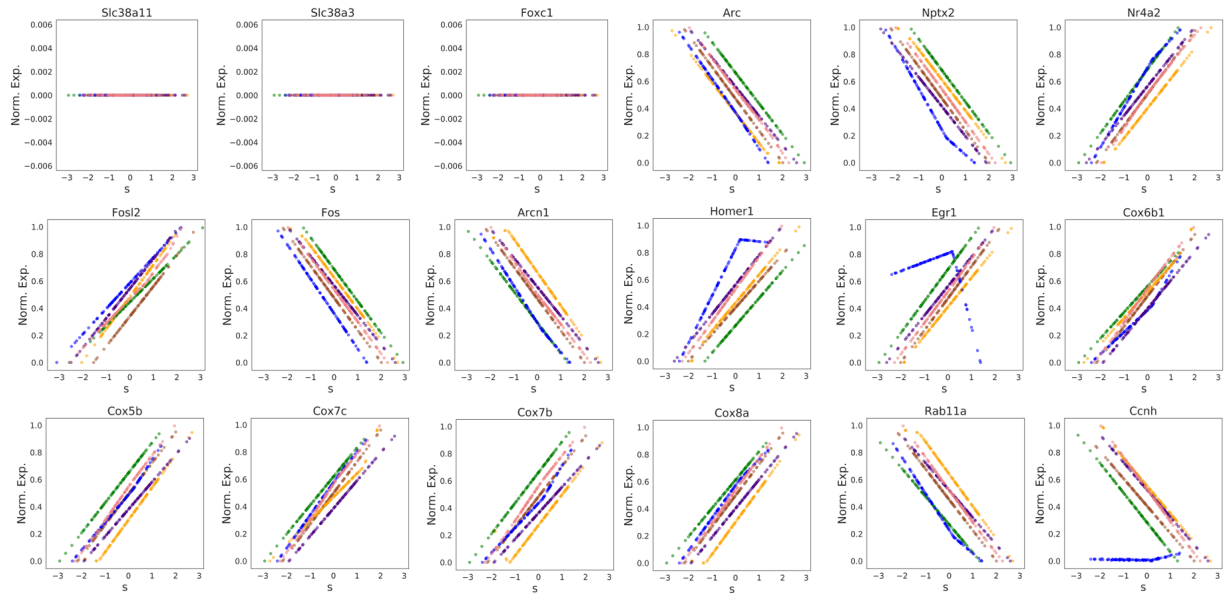


Figure S11: Consistency and interpretability of the continuous latent factor of cpl-mixVAE with 2 arms, for “L5 NP” in ALM. For each gene, we perform latent traversal analysis by fixing the discrete factor and changing the continuous variable. The y-axis corresponds to the normalized reconstructed gene expression, and the x-axis corresponds to the continuous variable. Colors denote the results for different runs.

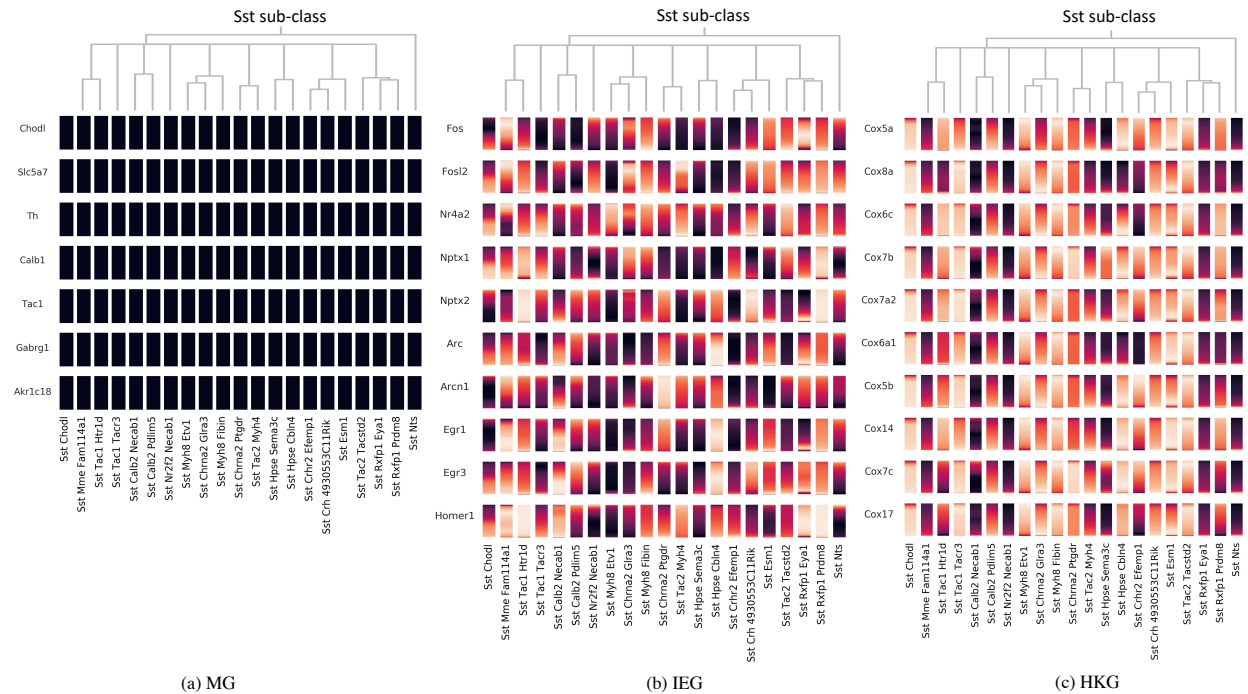
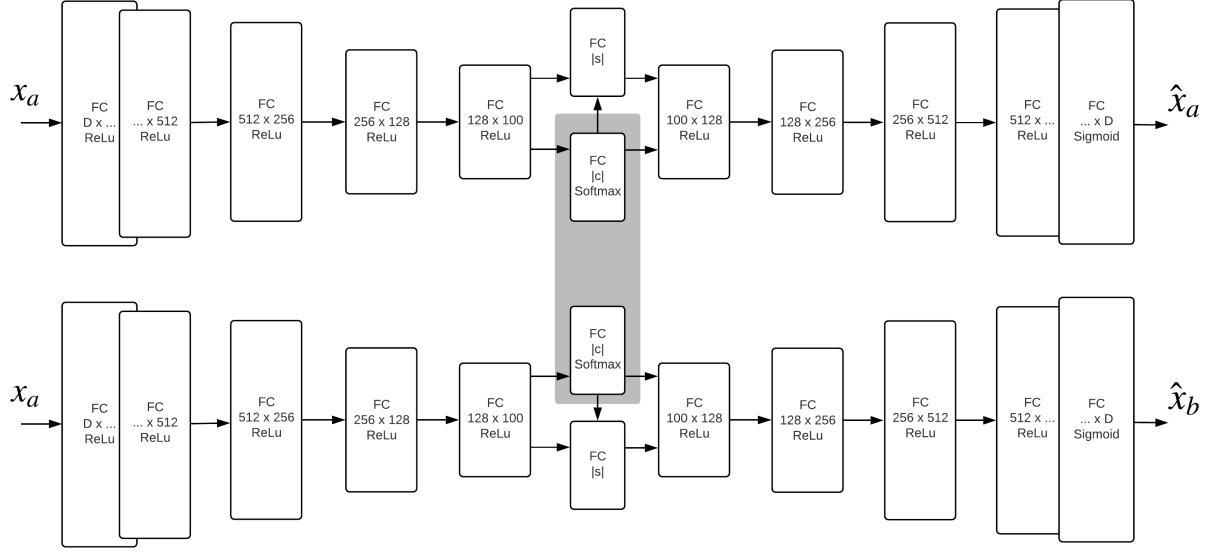
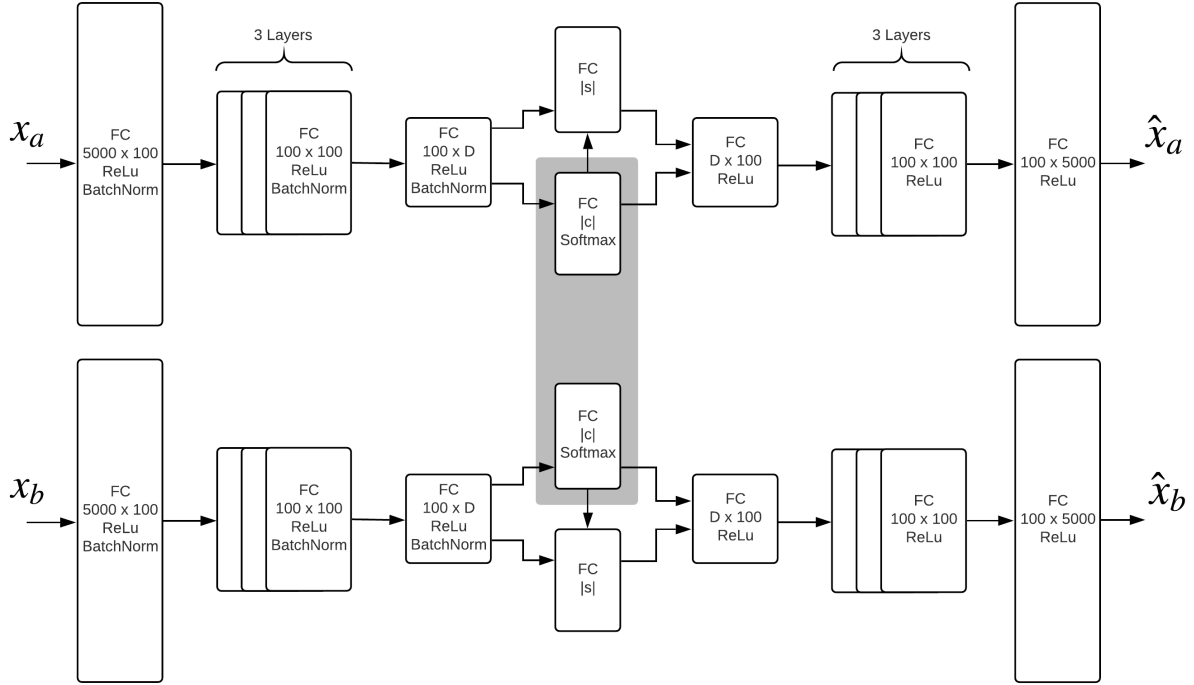


Figure S12: Continuous latent traversal analysis for 21 inhibitory cell types, i.e. Sst sub-class in ALM and VISp regions. Each of the 567 panels (rectangles) displays the traversal that is color-mapped to a normalized reconstructed gene expression value (colorbar) as a function of the state variable for different gene subsets: (a) marker genes (MG), (b) immediate early genes (IEG), and (c) housekeeping genes (HKG). Sst cell types on the x-axis are sorted based on a hierarchy (the dendrogram on top) suggested by Tasic et al. (2018).



(a) Benchmark datasets including MNIST and dSprites. The dimension of the input and first hidden layers depend on the image resolution i.e., D .



(b) scRNA-seq datasets

Figure S13: cpl-mixVAE architectures including 2 autoencoders.



Figure S14: Comparing the consensus discrete neuronal types obtained by cpl-mixVAE with the reference transcriptomic taxonomy in Tasic et al. 2018.



Figure S15: Cropped numbers in the SVHN dataset. At each panel, the first row shows the original image, and two bottom rows show reconstructed image by each mixture VAE in 2-arm cpl-mixVAE.