Appendix

In the appendix, we first show the ELBOs for the CVAEs under acyclic correlation graphs, as well as the derivation for the ELBO of $\text{CVAE}_{\text{corr}}$ (Eq. 8) for general correlation graphs. Also we show the counter example mentioned in Section 3.1. Then, we show the algorithm details for optimization with the loss function in Eq. 9 and the proofs of Theorems 2 and 3. At the end, we provide more information related to the models and inference settings.

A. ELBOs and Derivations

ELBO for CVAE_{ind} with acyclic graphs. CVAE_{ind} applies a correlated prior $p_0^{\text{corr}}(\boldsymbol{z})$ as in Eq. 4. Therefore,

$$\begin{aligned}
&\mathcal{L}^{\text{CVAE}_{\text{ind}}\text{-acyclic}}(\boldsymbol{\lambda}, \boldsymbol{\theta}) \\
=&\mathbb{E}_{q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right] - \text{KL}(q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})||p_0(\boldsymbol{z})) \\
=&\sum_{i=1}^{n}\Big(\mathbb{E}_{q_{\boldsymbol{\lambda}}(\boldsymbol{z}_i|\boldsymbol{x}_i)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z}_i)\right] - \text{KL}(q_{\boldsymbol{\lambda}}(\boldsymbol{z}_i|\boldsymbol{x}_i)||p_0(\boldsymbol{z}_i))\Big) \\
&+ \sum_{(v_i,v_j)\in E}\mathbb{E}_{q_{\boldsymbol{\lambda}}(\boldsymbol{z}_i|\boldsymbol{x}_i)q_{\boldsymbol{\lambda}}(\boldsymbol{z}_j|\boldsymbol{x}_j)}\log\frac{p_0(\boldsymbol{z}_i,\boldsymbol{z}_j)}{p_0(\boldsymbol{z}_i)p_0(\boldsymbol{z}_j)}.
\end{aligned} \tag{10}$$

This ELBO is a lower bound of the log probability $p(\boldsymbol{x};\boldsymbol{\theta})$ under the correlated prior (as in Eq. 4). Optimizing this ELBO helps the variational distributions to regularize according to the correlation graph G.

ELBO for $\text{CVAE}_{\text{corr}}$ with acyclic graphs. By changing q to be the correlated distribution as in Eq. 5, the ELBO in Eq. 10 becomes

$$\begin{aligned}
&\mathcal{L}^{\text{CVAE}_{\text{corr}}\text{-acyclic}}(\boldsymbol{\lambda}, \boldsymbol{\theta}) \\
=&\sum_{i=1}^{n}\Big(\mathbb{E}_{q_{\boldsymbol{\lambda}}(\boldsymbol{z}_i|\boldsymbol{x}_i)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z}_i)\right] - \text{KL}(q_{\boldsymbol{\lambda}}(\boldsymbol{z}_i|\boldsymbol{x}_i)||p_0(\boldsymbol{z}_i))\Big) \\
&- \sum_{(v_i,v_j)\in E}\Big(\text{KL}(q_{\boldsymbol{\lambda}}(\boldsymbol{z}_i,\boldsymbol{z}_j|\boldsymbol{x}_i,\boldsymbol{x}_j)||p_0(\boldsymbol{z}_i,\boldsymbol{z}_j)) \\
&- \text{KL}(q_{\boldsymbol{\lambda}}(\boldsymbol{z}_i|\boldsymbol{x}_i)||p_0(\boldsymbol{z}_i)) - \text{KL}(q_{\boldsymbol{\lambda}}(\boldsymbol{z}_j|\boldsymbol{x}_j)||p_0(\boldsymbol{z}_j))\Big).
\end{aligned} \tag{11}$$

Derivation for Eq. 8. We derive the ELBO of $\text{CVAE}_{\text{corr}}$ (Eq. 8) for general correlation graphs here. Recall that Eq. 7 shows that

$$\begin{aligned}
\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \geq \frac{1}{|\mathcal{A}_G|}\sum_{G'\in\mathcal{A}_G}\Big(&\mathbb{E}_{q_{\boldsymbol{\lambda}}^{G'}(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})] \\
&- \text{KL}(q_{\boldsymbol{\lambda}}^{G'}(\boldsymbol{z}|\boldsymbol{x})||p_0^{G'}(\boldsymbol{z}))\Big).
\end{aligned}$$

By the definition of $\mathcal{A}_G$, the right hand side of the above inequality equals the following sum

$$\begin{aligned}
&\sum_{i=1}^{n}\Big(\mathbb{E}_{q_{\boldsymbol{\lambda}}(\boldsymbol{z}_i|\boldsymbol{x}_i)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z}_i)\right] - \text{KL}(q_{\boldsymbol{\lambda}}(\boldsymbol{z}_i|\boldsymbol{x}_i)||p_0(\boldsymbol{z}_i))\Big) \\
&- \frac{1}{|\mathcal{A}_G|}\sum_{G'\in\mathcal{A}_G}\sum_{(v_i,v_j)\in E'}\Big(\text{KL}(q_{\boldsymbol{\lambda}}(\boldsymbol{z}_i,\boldsymbol{z}_j|\boldsymbol{x}_i,\boldsymbol{x}_j)||p_0(\boldsymbol{z}_i,\boldsymbol{z}_j)) \\
&- \text{KL}(q_{\boldsymbol{\lambda}}(\boldsymbol{z}_i|\boldsymbol{x}_i)||p_0(\boldsymbol{z}_i)) - \text{KL}(q_{\boldsymbol{\lambda}}(\boldsymbol{z}_j|\boldsymbol{x}_j)||p_0(\boldsymbol{z}_j))\Big)
\end{aligned}$$

The pairwise sum part of the above equation is an average over a sum over all edges of all maximal acyclic subgraphs of G. Therefore, for each edge $e = (v_i, v_j) \in E$, the number of times it appears in this pairwise sum part of the above sum is just the number of maximal acyclic subgraphs containing this edge. Therefore, this part can be viewed as a weighed sum over all edges in E, where the weights come from the fraction ratios in Definition 2. With this definition, we can further write the above sum as

$$\begin{aligned}
&\sum_{i=1}^{n}\Big(\mathbb{E}_{q_{\boldsymbol{\lambda}}(\boldsymbol{z}_i|\boldsymbol{x}_i)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z}_i)\right] - \text{KL}(q_{\boldsymbol{\lambda}}(\boldsymbol{z}_i|\boldsymbol{x}_i)||p_0(\boldsymbol{z}_i))\Big) \\
&- \sum_{(v_i,v_j)\in E} w^{\text{MAS}}_{G,(v_i,v_j)}\Big(\text{KL}(q_{\boldsymbol{\lambda}}(\boldsymbol{z}_i,\boldsymbol{z}_j|\boldsymbol{x}_i,\boldsymbol{x}_j)||p_0(\boldsymbol{z}_i,\boldsymbol{z}_j)) \\
&- \text{KL}(q_{\boldsymbol{\lambda}}(\boldsymbol{z}_i|\boldsymbol{x}_i)||p_0(\boldsymbol{z}_i)) - \text{KL}(q_{\boldsymbol{\lambda}}(\boldsymbol{z}_j|\boldsymbol{x}_j)||p_0(\boldsymbol{z}_j))\Big) \\
&= \mathcal{L}^{\text{CVAE}_{\text{corr}}}(\boldsymbol{\lambda}, \boldsymbol{\theta}),
\end{aligned}$$

which is exactly Eq. 8.

B. Counter example for Section 3.1

In Section 3.1, we mentioned that directly applying the ELBOs we derived (equations shown in Appendix A) will fail partly due to that optimizing over these ELBOs may lead the algorithm to learn useless parameters that lead the loss goes to infinity due to that these two equations are not guaranteed to be a lower bound of some log-likelihood function for general graphs. Here we provide a simple example to illustrate this.

Let us consider $G = (V, E)$ is a K_4 complete graph with $|V| = 4$ vertices and $\binom{|V|}{2} = 6$ edges. For simplicity, we consider the latent variables $z_1, z_2, z_3, z_4 \in \mathbb{R}$ and the prior distribution $p_0(z_i, z_j) = \mathcal{N}\left(\begin{pmatrix} z_i \\ z_j \end{pmatrix}; \boldsymbol{\mu} = \mathbf{0}_2, \Sigma = \begin{pmatrix} 1 & \tau \\ \tau & 1 \end{pmatrix}\right)$ for some $\tau \in (0, 1)$. If we extend the CVAE_{ind} and for the variational distribution, we set $q(z_i|\boldsymbol{x}_i)$ to be the normal distribution $N(\mu_i, \sigma_i^2)$, by

simply apply Eq. 10, then the loss function becomes

$$\mathcal{L} = \sum_{i=1}^{4} \Big(\mathbb{E}_{q_{\boldsymbol{\lambda}}(z_i|\boldsymbol{x}_i)} [\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|z_i)] + \mu_i^2 - \frac{1+2\tau^2}{2(1-\tau^2)}\sigma_i^2 + \ln(\sigma_i) \Big) - \frac{1}{2(1-\tau^2)} \sum_{1\le i<j\le 4} (\mu_i^2 + \mu_j^2 - 2\tau\mu_i\mu_j).$$

If we maintain σ_i's unchanged, set the model parameter $\boldsymbol{\theta}$ in a way that makes $p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|z_i)$ unrelated to z_i (e.g. set the parameter that multiply with z_i to be 0) and set $\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu$ and let $\mu \to \infty$, then $\mathcal{L}$ will go to $+\infty$ if $\tau > \frac{1}{2}$. Therefore, directly applying Eq. 10 does not work. Directly applying Eq. 11 (i.e. extending the CVAE$_{\text{corr}}$) will make the result even worse since it always has an optimal value at least as high as Eq. 10. In general, any general graphs with K_4 subgraph may suffer from the issue we just mentioned. We can not obtain useful latent embeddings by directly applying the Eqs. 10 and 11.

C. Algorithm details for optimization with the loss $\mathcal{L}^{\text{CVAE}_{\text{corr}}\text{-NS}}(\boldsymbol{\lambda}, \boldsymbol{\theta})$

We show the details of optimization with the loss $\mathcal{L}^{\text{CVAE}_{\text{corr}}\text{-NS}}(\boldsymbol{\lambda}, \boldsymbol{\theta})$ (Eq. 9) as in Algorithm 2.

Algorithm 2 Optimization with the loss $\mathcal{L}^{\text{CVAE}_{\text{corr}}\text{-NS}}(\boldsymbol{\lambda}, \boldsymbol{\theta})$

Input: undirected graph $G = (V = \{v_1, \dots, v_n\}, E)$, input data $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$, the parameter $\gamma > 0$.
Compute the edge weights $w_{G,e}^{\text{MAS}}$ using Algorithm 1.
Initialize the parameters $\boldsymbol{\lambda}$, $\boldsymbol{\theta}$.
while Not converged **do**
  Compute the gradient $\nabla_{\boldsymbol{\lambda},\boldsymbol{\theta}} \mathcal{L}^{\text{CVAE}_{\text{corr}}\text{-NS}}(\boldsymbol{\lambda}, \boldsymbol{\theta})$.
  Update the parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}$ using this gradient.
end while
Return The parameters $\boldsymbol{\lambda}$, $\boldsymbol{\theta}$.

If we subsample the vertices in V for the singleton part of this loss, subsample edges in E for the "positive" sample pairwise mart of this loss and subsample edges of the complete graph K_n for the "negative" sample pairwise part of this loss, then we get the stochastic optimization version for this algorithm.

D. Proofs

We prove the Theorems 2 and 3 here.

D.1. Proof of Theorem 2

Proof. Given the graph $G = (V, E)$ and the edge $(v_i, v_j) \in E$. Denote G's Laplacian matrix as L. Also denote $L_{-a,-b}$ as the sub-matrix of L after deleting the a^{th} row and the b^{th} column. Denote $L_{-ab,-cd}$ as the sub-matrix of L after deleting the a^{th}, b^{th} rows and the c^{th}, the d^{th} columns.

By Matrix Tree Theorem (Theorem 1 (Chaiken & Kleitman, 1978)), we know that the (i,i)-cofactor $C_{i,i} = |L_{-i,-i}|$ is the number of spanning trees of G.

Construct a graph $G' = (V, E/\{v_i, v_j\})$, i.e. the graph G after removing the edge (v_i, v_j). Denote the Laplacian matrix of G' as L'. Then we will find that the matrix $L'_{-i,-i}$ is the same with $L_{-i,-i}$ except that they $L_{-i,-i}$ is 1 larger than $L'_{-i,-i}$ on the entry at (j,j). By Matrix Tree Theorem, $|L'_{-i,-i}|$ is the number of spanning trees of G'. Since G differs from G' by only having one more edge (v_i, v_j), we know that $|L_{-i,-i}| - |L'_{-i,-i}|$ represents the number of spanning trees in G that contains the edge (v_i, v_j).

Denote the entry at (j,k) of $L_{-i,-i}$ as $L_{-i,-i;j,k}$. Since we know that

$$\begin{cases} |L_{-i,-i}| &= \sum_{k\neq i} (-1)^{j+k} L_{-i,-i;j,k} |L_{-ij,-ik}|, \\ |L'_{-i,-i}| &= \sum_{k\neq i} (-1)^{j+k} L'_{-i,-i;j,k} |L'_{-ij,-ik}|. \end{cases}$$

Subtract the second equation from the first one we get

$$|L_{-i,-i}| - |L'_{-i,-i}| = (-1)^{2j} |L_{-ij,-ij}|$$

which is just the complement of the Minor $M_{ij,ij}$ of L. Hence, the number of spanning trees of G that contains the edge (v_i, v_j) is $M_{ij,ij}$, the determinant of the sub-matrix of the Laplacian matrix L of G, after deleting the the i^{th}, j^{th} rows and the i^{th}, the j^{th} columns. □

D.2. Proof of Theorem 3

Proof. We borrow the notations from Appendix D.1. Given the undirected connected graph $G = (V, E)$ and an edge $(v_i, v_j) \in E$, we want to compute the ratio $w_{G,(v_i,v_j)}^{\text{MAS}}$. Since G is connected, this ratio is just the fraction of G's spanning trees containing the edge (v_i, v_j). By Theorem 1 and 2, this ratio is just $\frac{|L_{-ij,-ij}|}{|L_{-i,-i}|}$.

Since G is connected, it contains at least one spanning tree. Hence $|L_{-i,-i}| > 0$, which means $L_{-i,-i}$ is invertible. Therefore, we know that $\frac{|L_{-ij,-ij}|}{|L_{-i,-i}|} = L^{-1}_{-i,-i;j,j}$.

Consider the original Laplacian matrix L before deleting any row and column. Denote $|V| = n$. Since L is always symmetric and always have an eigenvector $\boldsymbol{v}_n = \frac{1}{\sqrt{n}}\mathbf{1}_n$ with corresponding eigenvalue $\lambda_n = 0$, we perform eigenvalue decomposition on L and write L as:

$$L = \sum_{k=1}^{n} \lambda_i \boldsymbol{v}_k \boldsymbol{v}_k^\top = \sum_{k=1}^{n-1} \lambda_i \boldsymbol{v}_k \boldsymbol{v}_k^\top.$$

Where $\lambda_1, \dots, \lambda_{n-1}, \lambda_n$ are L's eigenvalues and $\boldsymbol{v}_1, \dots, \boldsymbol{v}_{n-1}, \boldsymbol{v}_n$ are the corresponding orthogonal

unit eigenvectors (i.e. $Q_v = \begin{pmatrix} \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_{n-1} & \boldsymbol{v}_n \end{pmatrix}$ is an orthogonal matrix).

Denote $v_{a,b}$ as the b-the coordinate of $\boldsymbol{v}_a$ and $\boldsymbol{v}_{a,-b}$ as the sub-vector $\boldsymbol{v}_a$ after deleting the b^{th} coordinate. Then

$$L_{-i,-i} = \sum_{k=1}^{n-1} \lambda_k \boldsymbol{v}_{k,-i} \boldsymbol{v}_{k,-i}^\top.$$

$L_{-i,-i}$ is invertible, which is of rank $n-1$. While each of the matrix $\boldsymbol{v}_{k,-i}\boldsymbol{v}_{k,-i}^\top$ is of rank 1. Hence, we must have $\lambda_i \neq 0$ for all $i \in \{1, \ldots, n-1\}$.

Construct vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n \in \mathbb{R}^n$ such that

$$u_{k,a} = \begin{cases} v_{k,a} - v_{k,i} & \text{if } a \neq i \\ v_{k,a} & \text{if } a = i. \end{cases} \tag{12}$$

Also, construct the matrix

$$U = \sum_{k=1}^{n-1} \lambda_k^{-1} \boldsymbol{u}_k \boldsymbol{u}_k^\top.$$

Since we know that $\begin{pmatrix} \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_{n-1} & \boldsymbol{v}_n \end{pmatrix}$ forms an orthogonal basis and $\boldsymbol{v}_n = \frac{1}{\sqrt{n}}\mathbf{1}_n$, it is easy to see (after simple calculations) that

$$\boldsymbol{u}_{k,-i}\top \cdot \boldsymbol{v}_{k',-i} = \begin{cases} 1 & \text{if } k = k' \\ 0 & \text{if } k \neq k'. \end{cases}$$

Therefore, we will have

$$U_{-i,-i} L_{-i,-i} = I_{n-1}$$

which indicates that $U_{-i,-i} = L_{-i,-i}^{-1}$. Hence, the ratio we want to find is just $U_{-i,-i,j,j}$.

Recall the definition of $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$ in Eq. 12, denote $Q_u = \begin{pmatrix} \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_{n-1} & \boldsymbol{u}_n \end{pmatrix}$, we get $Q_u = P_i Q_v$, where

$$P_i = \begin{pmatrix} 1 & & & -1 & & & \\ & 1 & & -1 & & & \\ & & \ddots & \vdots & & & \\ & & & 1 & & & \\ & & & -1 & 1 & & \\ & & & \vdots & & \ddots & \\ & & & -1 & & & 1 \\ & & & -1 & & & \end{pmatrix} \in \mathbb{R}^{n \times n}$$

where the -1's appear on the i^{th} column. Hence, denote $D = \text{diag}(\lambda_1, \ldots, \lambda_n)$, we have

$$U = Q_u D^+ Q_u^\top = P_i Q_v D^+ Q_v^\top P_i^\top = P_i L^+ P_i^\top.$$

Therefore, the ratio $w^{\text{MAS}}_{G,(v_i,v_j)}$ we want to find, which is equal to $U_{-i,-i;j,j}$, is just the (j,j)-entry of $P_i L^+ P_i^\top$, which is

$$L^+_{i,i} - L^+_{i,j} - L^+_{j,i} + L^+_{j,j}.$$

□

E. Additional experiment settings

We provide some additional information related to the models and inference settings here.

For all methods, we set the latent embeddings to have the dimensionality $d = 100$.

For the standard VAEs, the CVAE$_{\text{ind}}$ and the CVAE$_{\text{corr}}$, we apply a two-layer feed-forward neural network for the generative model $p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z}_i)$ and a two-layer feed-forward neural network for the singleton variational approximations $q_{\boldsymbol{\lambda}}(\boldsymbol{z}_i|\boldsymbol{x}_i)$. The model likelihood functions $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$'s are Multinomial distribution (for the user matching and link prediction experiments) and Bernoulli distribution (for the spectral clustering experiments). The singleton variational approximations $q_{\boldsymbol{\lambda}}(\boldsymbol{z}_i|\boldsymbol{x}_i)$ are all diagonal normal distributions. The singleton prior density function $p_0(\cdot)$ is the standard normal distribution.

For the CVAEs, we set the pairwise prior density function $p_0(\cdot) = \mathcal{N}\left(\boldsymbol{\mu} = \mathbf{0}_{2d}, \Sigma = \begin{pmatrix} I_d & \tau \cdot I_d \\ \tau \cdot I_d & I_d \end{pmatrix}\right)$ for $\tau = 0.99$. It can be seen that, the singleton prior density function $p_0(\cdot)$ and the pairwise prior density function $p_0(\cdot,\cdot)$ satisfy the constraints in Eq. 2. For the CVAE$_{\text{corr}}$, we treat $q_{\boldsymbol{\lambda}}(\boldsymbol{z}_i, \boldsymbol{z}_j|\boldsymbol{x}_i, \boldsymbol{x}_j)$ as a multivariate normal distributions such that the covariance matrices $\text{Cov}(\boldsymbol{z}_i, \boldsymbol{z}_i)$, $\text{Cov}(\boldsymbol{z}_j, \boldsymbol{z}_j)$ and $\text{Cov}(\boldsymbol{z}_i, \boldsymbol{z}_j)$ are all diagonal matrices. Instead of only learning the singleton density function $q_{\boldsymbol{\lambda}}(\cdot|\cdot)$, the CVAE$_{\text{corr}}$ also learn a two-layer feedforward neural network that takes the concatenation $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ as input and output the covariance between $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ on each of these d dimensions. As a result, $q_{\boldsymbol{\lambda}}(\boldsymbol{z}_i, \boldsymbol{z}_j|\boldsymbol{x}_i, \boldsymbol{x}_j)$ can be factorized as a product of d bi-variate normal distributions, whose marginal distributions on $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ are consistent with the singleton variational approximations $q_{\boldsymbol{\lambda}}(\boldsymbol{z}_i|\boldsymbol{x}_i)$ and $q_{\boldsymbol{\lambda}}(\boldsymbol{z}_j|\boldsymbol{x}_j)$, respectively.

For GraphSAGE, we choose to use $K = 2$ aggregation steps and use the mean aggregator function. We use $Q = 20$ negative samples to optimize the loss function.

For all methods, we apply stochastic gradient optimizations with a step size of 10^{-3}. We use the Adam algorithm (Kingma & Ba, 2015) to adjust the learning rates. All methods involve with stochastic batches with singleton terms. For these terms, we use a batch size $B_1 = 64$. For the CVAEs, there are some pairwise terms of the sum involve sampling edges from the graph $G = (V, E)$ (i.e. the "positive sampling") or sampling edges from the complete graph K_n (i.e. the "negative sampling"). We use a batch size $B_2 = 256$ for sampling these pairwise terms.