# Appendix for Bayesian Active Causal Discovery with Multi-Fidelity Experiments

**Anonymous Author(s)**
Affiliation
Address
email

## Contents



Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

# A Monte Carlo Approximation for $f(j, v, m)$

## A.1 Derivation Process for $f(j, v, m)$

Considering that the mutual information is not directly tractable, we approximate $f(j, v, m)$ by:

$$
\begin{aligned}
f(j, v, m) = & -\frac{1}{\lambda_m \cdot K_1 \cdot L_1} \sum_{k_1=1}^{K_1} \sum_{l_1=1}^{L_1} \log \left[ \frac{1}{C} \sum_{c_1=1}^{C_1} p(\boldsymbol{x}_m^{(k_1,l_1)} | \boldsymbol{\phi}_m^{(c_1)}, \boldsymbol{e}) \right] \\
& + \frac{1}{\lambda_m \cdot K_2 \cdot L_2 \cdot C_2} \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{L_2} \sum_{c_2=1}^{C_2} \log \left[ p(\boldsymbol{x}_m^{(k_2,l_2,c_2)} | \boldsymbol{\phi}_M^{(k_2)}, \boldsymbol{e}) \right],
\end{aligned}
$$

where $\boldsymbol{e} = \{(j, v), m\}$ is the experiment to be designed, $\boldsymbol{\phi}_m^{(c_1)}, \boldsymbol{\phi}_m^{(k_1)} \sim p(\boldsymbol{\phi}_m | D)$, $\boldsymbol{x}_m^{(k_1,l_1)} \sim p(\boldsymbol{x} | \boldsymbol{\phi}_m^{(k_1)}, \boldsymbol{e})$, $\boldsymbol{\phi}_M^{(k_2)} \sim p(\boldsymbol{\phi}_M | D)$, $\boldsymbol{\phi}_m^{(k_2,l_2)} \sim p(\boldsymbol{\phi}_m | \boldsymbol{\phi}_M^{(k_2)}, D)$ and $\boldsymbol{x}_m^{(k_2,l_2,c_2)} \sim p(\boldsymbol{x} | \boldsymbol{\phi}_m^{(k_2,l_2)}, \boldsymbol{e})$.

We present the detailed approximation process as follows:

$$
\begin{aligned}
f(j, v, m) &= \frac{1}{\lambda_m} I(\boldsymbol{x}; \boldsymbol{\phi}_M | \boldsymbol{e}, D) \\
&= \frac{1}{\lambda_m} \left[ H(\boldsymbol{x} | \boldsymbol{e}, D) - H(\boldsymbol{x} | \boldsymbol{\phi}_M, \boldsymbol{e}, D) \right] \\
&= \frac{1}{\lambda_m} \left[ -\mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{e},D)} \left[ \log p(\boldsymbol{x} | \boldsymbol{e}, D) + \mathbb{E}_{p(\boldsymbol{\phi}_M|D)} \left[ \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{\phi}_M,\boldsymbol{e})} \left[ \log p(\boldsymbol{x} | \boldsymbol{e}, \boldsymbol{\phi}_M) \right] \right] \right] \right] \\
&= \underbrace{\frac{1}{\lambda_m} \left[ -\mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{e},D)} \left[ \log \mathbb{E}_{p(\boldsymbol{\phi}_m|\boldsymbol{e},D)} \left[ p(\boldsymbol{x} | \boldsymbol{e}, \boldsymbol{\phi}_m) \right] \right] \right]}_{E} \\
&\quad + \underbrace{\frac{1}{\lambda_m} \left[ \mathbb{E}_{p(\boldsymbol{\phi}_M|D)} \left[ \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{e},\boldsymbol{\phi}_M)} \left[ \log p(\boldsymbol{x} | \boldsymbol{e}, \boldsymbol{\phi}_M) \right] \right] \right]}_{F}
\end{aligned}
$$

For part $E$, we can estimate it by

$$
E = -\frac{1}{\lambda_m \cdot K_1 \cdot L_1} \sum_{k_1=1}^{K_1} \sum_{l_1=1}^{L_1} \log \left[ \frac{1}{C} \sum_{c_1=1}^{C_1} p(\boldsymbol{x}_m^{(k_1,l_1)} | \boldsymbol{\phi}_m^{(c_1)}, \boldsymbol{e}) \right],
$$

where for the first expectation on $p(\boldsymbol{\phi}_m | \boldsymbol{e}, D)$, we first sample $\boldsymbol{\phi}_m^{(k_1)}$ from $\boldsymbol{\phi}_m^{(k_1)} \sim p(\boldsymbol{\phi}_m | \boldsymbol{e}, D)$ for $K_1$ times, and then for each $\boldsymbol{\phi}_m^{(k_1)}$, we sample $\boldsymbol{x}_m^{(k_1,l_1)}$ from $\boldsymbol{x}_m^{(k_1,l_1)} \sim p(\boldsymbol{x} | \boldsymbol{\phi}_m^{(k_1)}, \boldsymbol{e})$ for $L_1$ times. For the second expectation on $p(\boldsymbol{\phi}_m | \boldsymbol{e}, D)$, we sample $\boldsymbol{\phi}_m^{(c_1)} \sim p(\boldsymbol{\phi}_m | \boldsymbol{e}, D)$ for $C_1$ times.

For part $F$, we have

$$
\begin{aligned}
F &= \frac{1}{\lambda_m} \cdot \left[ \mathbb{E}_{p(\boldsymbol{\phi}_M|D)} \left[ \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{\phi}_M,\boldsymbol{e})} \left[ \log p(\boldsymbol{x} | \boldsymbol{\phi}_M, \boldsymbol{e}) \right] \right] \right] \\
&= \frac{1}{\lambda_m} \cdot \left[ \mathbb{E}_{p(\boldsymbol{\phi}_M|D)} \left[ \int p(\boldsymbol{x} | \boldsymbol{\phi}_M, \boldsymbol{e}) \log p(\boldsymbol{x} | \boldsymbol{\phi}_M, \boldsymbol{e}) \, d\boldsymbol{x} \right] \right] \\
&= \frac{1}{\lambda_m} \cdot \left[ \mathbb{E}_{p(\boldsymbol{\phi}_M|D)} \left[ \int \int p(\boldsymbol{x} | \boldsymbol{\phi}_m, \boldsymbol{e}) p(\boldsymbol{\phi}_m | \boldsymbol{\phi}_M) \, d\boldsymbol{\phi}_m \log p(\boldsymbol{x} | \boldsymbol{\phi}_M, \boldsymbol{e}) \, d\boldsymbol{x} \right] \right] \\
&= \frac{1}{\lambda_m} \cdot \left[ \mathbb{E}_{p(\boldsymbol{\phi}_M|D,\boldsymbol{e})} \left[ \int \int p(\boldsymbol{x} | \boldsymbol{\phi}_m, \boldsymbol{e}) p(\boldsymbol{\phi}_m | \boldsymbol{\phi}_M) \log p(\boldsymbol{x} | \boldsymbol{\phi}_M, \boldsymbol{e}) \, d\boldsymbol{x} \, d\boldsymbol{\phi}_m \right] \right] \\
&= \frac{1}{\lambda_m} \cdot \left[ \mathbb{E}_{p(\boldsymbol{\phi}_M|D,\boldsymbol{e})} \left[ \int \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{\phi}_m,\boldsymbol{e})} \left[ p(\boldsymbol{\phi}_m | \boldsymbol{\phi}_M) \log p(\boldsymbol{x} | \boldsymbol{\phi}_M, \boldsymbol{e}) \right] \, d\boldsymbol{\phi}_m \right] \right] \\
&= \frac{1}{\lambda_m} \cdot \left[ \mathbb{E}_{p(\boldsymbol{\phi}_M|D,\boldsymbol{e})} \left[ \mathbb{E}_{p(\boldsymbol{\phi}_m|\boldsymbol{\phi}_M)} \left[ \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{\phi}_m,\boldsymbol{e})} \left[ \log p(\boldsymbol{x} | \boldsymbol{\phi}_M, \boldsymbol{e}) \right] \right] \right] \right].
\end{aligned}
$$

It can be estimated by

$$
\frac{1}{\lambda_m \cdot K_2 \cdot L_2 \cdot C_2} \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{L_2} \sum_{c_2=1}^{C_2} \log \left[ p(\boldsymbol{x}_m^{(k_2,l_2,c_2)} | \boldsymbol{\phi}_M^{(k_2)}, \boldsymbol{e}) \right],
$$

where for the expectation on $p(\boldsymbol{\phi}_M|D,\boldsymbol{e})$, we sample $\boldsymbol{\phi}_M^{(k_2)}$ from $\boldsymbol{\phi}_M^{(k_2)} \sim p(\boldsymbol{\phi}_M|\boldsymbol{e},D)$ for $K_2$ times. For the expectation on $p(\boldsymbol{\phi}_m|\boldsymbol{\phi}_M)$, for each $\boldsymbol{\phi}_M^{(k_2)}$, we sample $\boldsymbol{\phi}_m^{(k_2,l_2)}$ from $\boldsymbol{\phi}_m^{(k_2,l_2)} \sim p(\boldsymbol{\phi}_m|\boldsymbol{\phi}_M^{(k_2)})$ for $L_2$ times. For the expectation on $p(\boldsymbol{x}|\boldsymbol{\phi}_m,\boldsymbol{e})$, for each $\boldsymbol{\phi}_M^{(k_2)}$ and $\boldsymbol{\phi}_m^{(k_2,l_2)}$, we sample $\boldsymbol{x}_m^{(k_2,l_2,c_2)}$ from $\boldsymbol{x}_m^{(k_2,l_2,c_2)} \sim p(\boldsymbol{x}|\boldsymbol{\phi}_m^{(k_2,l_2)},\boldsymbol{e})$ for $C_2$ times.

Therefore, we can conclude that $f(j,v,m)$ can be estimated by

$$
\begin{aligned}
f(j,v,m) = &-\frac{1}{\lambda_m \cdot K_1 \cdot L_1} \sum_{k_1=1}^{K_1} \sum_{l_1=1}^{L_1} \log \left[ \frac{1}{C} \sum_{c_1=1}^{C_1} p(\boldsymbol{x}_m^{(k_1,l_1)}|\boldsymbol{\phi}_m^{(c_1)},\boldsymbol{e}) \right] \\
&+ \frac{1}{\lambda_m \cdot K_2 \cdot L_2 \cdot C_2} \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{L_2} \sum_{c_2=1}^{C_2} \log \left[ p(\boldsymbol{x}_m^{(k_2,l_2,c_2)}|\boldsymbol{\phi}_M^{(k_2)},\boldsymbol{e}) \right],
\end{aligned}
$$

where $\boldsymbol{\phi}_m^{(c_1)}, \boldsymbol{\phi}_m^{(k_1)} \sim p(\boldsymbol{\phi}_m|D)$, $\boldsymbol{x}_m^{(k_1,l_1)} \sim p(\boldsymbol{x}|\boldsymbol{\phi}_m^{(k_1)},\boldsymbol{e})$, $\boldsymbol{\phi}_M^{(k_2)} \sim p(\boldsymbol{\phi}_M|D)$, $\boldsymbol{\phi}_m^{(k_2,l_2)} \sim p(\boldsymbol{\phi}_m|\boldsymbol{\phi}_M^{(k_2)},D)$ and $\boldsymbol{x}_m^{(k_2,l_2,c_2)} \sim p(\boldsymbol{x}|\boldsymbol{\phi}_m^{(k_2,l_2)},\boldsymbol{e})$.

Obviously, the above approximation of $f(j,v,m)$ only depends on $p(\boldsymbol{\phi}_m|D)$, $p(\boldsymbol{\phi}_m|\boldsymbol{\phi}_M,D)$ and $p(\boldsymbol{x}|\boldsymbol{\phi}_m,\boldsymbol{e})$. In the next, we show how to sample from them in Section A.2, A.3 and A.4, respectively.

## A.2 Sampling from $p(\boldsymbol{\phi}_m|D)$

Basically, sampling from the posterior of "$p(\cdot|D)$" is not easy. To solve this problem, as mentioned in the main paper, we introduce a variational probability "$q$" to approximate "$p$". In specific, in order to sample from $p(\boldsymbol{\phi}_m|D)$, we first obtain a sample $\boldsymbol{\phi}_1$ from $\boldsymbol{\phi}_1 \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{I})$, and then get $\boldsymbol{\phi}_m$ from the distribution $q(\boldsymbol{\phi}_m|\boldsymbol{\phi}_1)$.

Since

$$
q(\boldsymbol{\phi}_m|\boldsymbol{\phi}_1) = \int_{\boldsymbol{\phi}_{m-1}} \cdots \int_{\boldsymbol{\phi}_2} q(\boldsymbol{\phi}_m,\boldsymbol{\phi}_{m-1},\ldots,\boldsymbol{\phi}_2|\boldsymbol{\phi}_1)\, d\boldsymbol{\phi}_{m-1} \ \ldots \ d\boldsymbol{\phi}_2.
$$

and

$$
\begin{aligned}
q(\boldsymbol{\phi}_m,\boldsymbol{\phi}_{m-1},\ldots,\boldsymbol{\phi}_2|\boldsymbol{\phi}_1) &= \prod_{i=2}^{m} q(\boldsymbol{\phi}_i|\boldsymbol{\phi}_{i-1}) \\
q(\boldsymbol{\phi}_i|\boldsymbol{\phi}_{i-1}) &= \mathcal{N}(\boldsymbol{c}_i\boldsymbol{\phi}_{i-1}+\boldsymbol{d}_i, \boldsymbol{\sigma}_i^2\boldsymbol{I}),
\end{aligned}
$$

we have $q(\boldsymbol{\phi}_m|\boldsymbol{\phi}_1)$ is a Gaussian distribution, which is easy for sampling.

In our model, $\boldsymbol{c}_i$ and $\boldsymbol{\sigma}_i^2\boldsymbol{I}$ are diagonal matrices, which means that the dimensions in $\boldsymbol{\phi}_i$ are independent with each other. We denote $\boldsymbol{\phi}_i = [\phi_{i,1},\phi_{i,2}\ldots\phi_{i,d}]$, where $\phi_{i,j}$ is the $j$th element of $\boldsymbol{\phi}_i$. Then, we have

$$
q(\boldsymbol{\phi}_m,\boldsymbol{\phi}_{m-1},\ldots,\boldsymbol{\phi}_2|\boldsymbol{\phi}_1) = \prod_{j=1}^{d} q(\phi_{m,j},\phi_{m-1,j},\ldots,\phi_{2,j}|\phi_{1,j}).
$$

So our target can be converted to calculate the probability $q(\phi_{m,j},\phi_{m-1,j},\ldots,\phi_{2,j}|\phi_{1,j})$ for all dimensions $\forall 1 \le j \le d$. Let $c_{i,j}, d_{i,j}$ and $\sigma_{i,j}^2$ be the $j$th element of $\boldsymbol{c}_i$, $\boldsymbol{d}_i$ and $\boldsymbol{\sigma}_i^2$, respectvely. We assume that $\sigma_{i,j} = \sqrt{c_{i-1,j}^2+1}\cdot\sigma_{i-1,j}$ $(i \ge 4)$ and $\sigma_{3,j} = \sigma_{2,j} = e$, where $e$ is the hyper-parameter. Suppose $\mu_{i,j}$ is the mean of the Gaussian distribution for $q(\phi_{i,j}|\phi_{i-1,j})$, that is,

$$
\mu_{i,j} = c_{i,j}\phi_{i-1,j} + d_{i,j} \ (i \ge 2),
$$

then, the approximated joint distribution can be represented as

$$
\begin{aligned}
q(\phi_{m,j},\phi_{m-1,j},\ldots,\phi_{2,j}|\phi_{1,j}) &= \prod_{i=2}^{m} q(\phi_{i,j}|\phi_{i-1,j}) \\
&= \prod_{i=2}^{m} \frac{1}{\sqrt{2\pi}\sigma_{i,j}} \cdot e^{\frac{-1}{2\sigma_{i,j}^2}(\phi_{i,j}-\mu_{i,j})^2}.
\end{aligned}
$$

Then, we integrate $\phi_{2,j}, \phi_{3,j}, \dots, \phi_{m-1,j}$ sequentially to obtain $q(\phi_{m,j}|\phi_{1,j})$. First of all, we integrate $\phi_{2,j}$ for the joint distribution, where we have:

$$
\begin{aligned}
& q(\phi_{m,j}, \phi_{m-1,j}, \dots, \phi_{3,j}|\phi_{1,j}) \\
=& \int q(\phi_{m,j}, \phi_{m-1,j}, \dots, \phi_{3,j}, \phi_{2,j}|\phi_{1,j})\, d\phi_{2,j} \\
=& \int \prod_{i=2}^{m} \frac{1}{\sqrt{2\pi}\sigma_{i,j}} \cdot e^{\frac{-1}{2\sigma_{i,j}^2}(\phi_{i,j}-\mu_{i,j})^2}\, d\phi_{2,j} \\
=& \prod_{i=4}^{m} \frac{1}{\sqrt{2\pi}\sigma_{i,j}} \cdot e^{\frac{-1}{2\sigma_{i,j}^2}(\phi_{i,j}-\mu_{i,j})^2} \cdot \frac{1}{\sqrt{2\pi}\sigma_{3,j}} \cdot \frac{1}{\sqrt{2\pi}\sigma_{2,j}} \cdot \int e^{\frac{-1}{2\sigma_{3,j}^2}[\phi_{3,j}-(c_{3,j}\phi_{2,j}+d_{3,j})]^2} \cdot \\
& e^{\frac{-1}{2\sigma_{2,j}^2}[\phi_{2,j}-(w_2\phi_{1,j}+d_{3,j})]^2}\, d\phi_{2,j}.
\end{aligned}
$$

Denote $\bar{c}_{2,j} = c_{2,j}$ and $\bar{d}_{2,j} = d_{2,j}$, and because of $\sigma_{3,j} = \sigma_{2,j}$, we have

$$
\begin{aligned}
& q(\phi_{m,j}, \phi_{m-1,j}, \dots, \phi_{3,j}|\phi_{1,j}) \\
=& \frac{1}{\sqrt{2\pi}\sigma_{2,j}} \cdot \prod_{i=4}^{m} \frac{1}{\sqrt{2\pi}\sigma_{i,j}} \cdot e^{\frac{-1}{2\sigma_{i,j}^2}(\phi_{i,j}-\mu_{i,j})^2} \cdot \frac{1}{\sqrt{2\pi}\sigma_{3,j}} \int e^{\frac{-1}{2\sigma_{3,j}^2}[\phi_{3,j}-(c_{3,j}\phi_{2,j}+d_{3,j})]^2} \cdot \\
& e^{\frac{-1}{2\sigma_{3,j}^2}\left[\phi_{2,j}-(\bar{c}_{2,j}\phi_{1,j}+\bar{d}_{2,j})\right]^2}\, d\phi_{2,j} \\
=& \frac{1}{\sqrt{2\pi}\sigma_{2,j}} \cdot \prod_{i=4}^{m} \frac{1}{\sqrt{2\pi}\sigma_{i,j}} \cdot e^{\frac{-1}{2\sigma_{i,j}^2}(\phi_{i,j}-\mu_{i,j})^2} \cdot \frac{1}{\sqrt{2\pi}\sigma_{3,j}} \int \cdot \\
& e^{\frac{-1}{2\sigma_{3,j}^2}\underbrace{\left\{[\phi_{3,j}-(c_{3,j}\phi_{2,j}+d_{3,j})]^2+\left[\phi_{2,j}-(\bar{c}_{2,j}\phi_{1,j}+\bar{d}_{2,j})\right]^2\right\}}_{S_1}}\, d\phi_{2,j}.
\end{aligned}
$$

For $S_1$, we have

$$
\begin{aligned}
S_1 =& [\phi_{3,j}-(c_{3,j}\phi_{2,j}+d_{3,j})]^2+\left[\phi_{2,j}-(\bar{c}_{2,j}\phi_{1,j}+\bar{d}_{2,j})\right]^2 \\
=& \phi_{3,j}^2 + c_{3,j}^2\phi_{2,j}^2 + d_{3,j}^2 + 2d_{3,j}c_{3,j}\phi_{2,j} - 2c_{3,j}\phi_{3,j}\phi_{2,j} - 2d_{3,j}\phi_{3,j} + \phi_{2,j}^2 \\
& + \bar{c}_{2,j}^2\phi_{1,j}^2 + \bar{d}_{2,j}^2 + 2\bar{d}_{2,j}\bar{c}_{2,j}\phi_{1,j} - 2\bar{c}_{2,j}\phi_{1,j} - 2d_{3,j}\phi_{3,j} \\
=& (c_{3,j}^2+1) \cdot \\
& \left[\phi_{2,j}^2 + \frac{2(d_{3,j}c_{3,j} - c_{3,j}\phi_{3,j} - \bar{c}_{2,j}\phi_{1,j} - \bar{d}_{2,j})}{c_{3,j}^2+1} + \left(\frac{d_{3,j}c_{3,j} - c_{3,j}\phi_{3,j} - \bar{c}_{2,j}\phi_{1,j} - \bar{d}_{2,j}}{c_{3,j}^2+1}\right)^2\right] \\
& - \frac{(d_{3,j}c_{3,j} - c_{3,j}\phi_{3,j} - \bar{c}_{2,j}\phi_{1,j} - \bar{d}_{2,j})^2}{c_{3,j}^2+1} + \phi_{3,j}^2 + \bar{c}_{2,j}^2\phi_{1,j}^2 + \bar{d}_{2,j}^2 + 2\bar{d}_{2,j}\bar{c}_{2,j}\phi_{1,j} - 2d_{3,j}\phi_{3,j} \\
=& (c_{3,j}^2+1) \cdot \left(\phi_{2,j} - \frac{c_{3,j}\phi_{3,j} + \bar{c}_{2,j}\phi_{1,j} + \bar{d}_{2,j} - d_{3,j}c_{3,j}}{c_{3,j}^2+1}\right)^2 \\
& + \frac{c_{3,j}^2\phi_{3,j}^2 + \bar{c}_{2,j}^2c_{3,j}^2\phi_{1,j}^2 + \bar{d}_{2,j}^2c_{3,j}^2 + 2d_{3,j}\bar{c}_{2,j}c_{3,j}\phi_{1,j} - \bar{c}_{2,j}^2\phi_{1,j}^2 - \bar{d}_{2,j}^2 - 2\bar{d}_{2,j}\bar{c}_{2,j}\phi_{1,j}}{c_{3,j}^2+1} \\
& + \frac{\phi_{3,j}^2 + \bar{c}_{2,j}^2\phi_{1,j}^2 + \bar{d}_{2,j}^2 + 2\bar{d}_{2,j}\bar{c}_{2,j}\phi_{1,j} - 2d_{3,j}\phi_{3,j} - c_{3,j}^2d_{3,j}^2 - c_{3,j}^2\phi_{3,j}^2 + 2c_{3,j}^2d_{3,j}\phi_{3,j}}{c_{3,j}^2+1} \\
& + \frac{2c_{3,j}\bar{d}_{2,j}d_{3,j} - 2c_{3,j}\phi_{3,j}\bar{c}_{2,j}\phi_{1,j} - 2c_{3,j}\phi_{3,j}\bar{d}_{2,j} + 2\bar{d}_{2,j}\bar{c}_{2,j}c_{3,j}^2\phi_{1,j} - 2d_{3,j}c_{3,j}^2\phi_{3,j}}{c_{3,j}^2+1}.
\end{aligned}
$$

Then we have

$$
\begin{aligned}
S_1 =&(c_{3,j}^2+1)\cdot\left(\phi_{2,j}-\frac{c_{3,j}\phi_{3,j}+\bar{c}_{2,j}\phi_{1,j}+\bar{d}_{2,j}-d_{3,j}c_{3,j}}{c_{3,j}^2+1}\right)^2\\
&+\frac{\phi_{3,j}^2-2(\bar{c}_{2,j}c_{3,j}\phi_{1,j}+\bar{d}_{2,j}c_{3,j}+d_{3,j})\phi_{3,j}}{c_{3,j}^2+1}\\
&+\frac{\left(\bar{c}_{2,j}c_{3,j}\phi_{1,j}+\bar{d}_{2,j}c_{3,j}+d_{3,j}\right)^2-d_{3,j}^2\cdot(c_{3,j}^2+1)}{c_{3,j}^2+1}\\
=&(c_{3,j}^2+1)\cdot\left(\phi_{2,j}-\frac{c_{3,j}\phi_{3,j}+\bar{c}_{2,j}\phi_{1,j}+\bar{d}_{2,j}-d_{3,j}c_{3,j}}{c_{3,j}^2+1}\right)^2\\
&+\frac{1}{c_{3,j}^2+1}\cdot\left[\phi_{3,j}-(c_{3,j}\bar{c}_{2,j}\phi_{1,j}+\bar{d}_{2,j}c_{3,j}+d_{3,j})\right]^2-d_{3,j}^2.
\end{aligned}
$$

Therefore we have

$$
\begin{aligned}
&q(\phi_{m,j},\phi_{m-1,j},\dots,\phi_{3,j}|\phi_{1,j})\\
=&\frac{1}{\sqrt{2\pi}\sigma_{2,j}}\cdot\prod_{i=4}^{m}\frac{1}{\sqrt{2\pi}\sigma_{i,j}}\cdot e^{\frac{-1}{2\sigma_{i,j}^2}(\phi_{i,j}-\mu_{i,j})^2}\cdot\\
&\underbrace{\left[\int\frac{\sqrt{c_{3,j}^2+1}}{\sqrt{2\pi}\sigma_{3,j}}e^{\frac{-(c_{3,j}^2+1)}{2\sigma_{3,j}^2}\left(\phi_{2,j}-\frac{c_{3,j}\phi_{3,j}+\bar{c}_{2,j}\phi_{1,j}+\bar{d}_{2,j}-d_{3,j}c_{3,j}}{c_{3,j}^2+1}\right)^2}d\phi_{2,j}\right]}_{S_2}\cdot\\
&e^{\frac{-1}{2\sigma_{3,j}^2\cdot(c_{3,j}^2+1)}\left[\phi_{3,j}-(c_{3,j}\bar{c}_{2,j}\phi_{1,j}+\bar{d}_{2,j}c_{3,j}+d_{3,j})\right]^2}\cdot e^{\frac{d_{3,j}^2}{2\sigma_{3,j}^2}}\cdot\frac{1}{\sqrt{c_{3,j}^2+1}}
\end{aligned}
$$

The $S_2$ part is the integration form of $\phi_{2,j}\sim\mathcal{N}(\frac{c_{3,j}\phi_{3,j}+\bar{c}_{2,j}\phi_{1,j}+\bar{d}_{2,j}-d_{3,j}c_{3,j}}{c_{3,j}^2+1},\frac{\sigma_{3,j}^2}{c_{3,j}^2+1})$, which is equal to 1, so we have

$$
\begin{aligned}
&q(\phi_{m,j},\phi_{m-1,j},\dots,\phi_{3,j}|\phi_{1,j})\\
=&\frac{1}{\sqrt{2\pi}\sigma_{2,j}}\cdot\prod_{i=4}^{m}\frac{1}{\sqrt{2\pi}\sigma_{i,j}}\cdot e^{\frac{-1}{2\sigma_{i,j}^2}(\phi_{i,j}-\mu_{i,j})^2}\cdot e^{\frac{-1}{2\sigma_{3,j}^2\cdot(c_{3,j}^2+1)}\left[\phi_{3,j}-(c_{3,j}\bar{c}_{2,j}\phi_{1,j}+\bar{d}_{2,j}c_{3,j}+d_{3,j})\right]^2}\cdot\\
&e^{\frac{d_{3,j}^2}{2\sigma_{3,j}^2}}\cdot\frac{1}{\sqrt{c_{3,j}^2+1}}.
\end{aligned}
$$

We denote $\bar{c}_{3,j}=c_{3,j}\bar{c}_{2,j}$ and $\bar{d}_{3,j}=\bar{d}_{2,j}c_{3,j}+d_{3,j}$, and denote $r_{2,j}=e^{\frac{d_{3,j}^2}{2\sigma_{3,j}^2}}\cdot\frac{1}{\sqrt{c_{3,j}^2+1}}$, so we have

$$
\begin{aligned}
&q(\phi_{m,j},\phi_{m-1,j},\dots,\phi_{3,j}|\phi_{1,j})\\
=&\frac{1}{\sqrt{2\pi}\sigma_{2,j}}\cdot\prod_{i=4}^{m}\frac{1}{\sqrt{2\pi}\sigma_{i,j}}\cdot e^{\frac{-1}{2\sigma_{i,j}^2}(\phi_{i,j}-\mu_{i,j})^2}\cdot e^{\frac{-1}{2\sigma_{3,j}^2\cdot(c_{3,j}^2+1)}\left[\phi_{3,j}-(\bar{c}_{3,j}\phi_{1,j}+\bar{d}_{3,j})\right]^2}\cdot r_{2,j}\\
=&\frac{1}{\sqrt{2\pi}\sigma_{2,j}}\cdot\prod_{i=5}^{m}\frac{1}{\sqrt{2\pi}\sigma_{i,j}}\cdot e^{\frac{-1}{2\sigma_{i,j}^2}(\phi_{i,j}-\mu_{i,j})^2}\cdot\frac{1}{\sqrt{2\pi}\sigma_4}\cdot e^{\frac{-1}{2\sigma_4^2}\left[\phi_4-(c_4\phi_{3,j}+d_4)\right]^2}\cdot\\
&e^{\frac{-1}{2\sigma_{3,j}^2\cdot(c_{3,j}^2+1)}\left[\phi_{3,j}-(\bar{c}_{3,j}\phi_{1,j}+\bar{d}_{3,j})\right]^2}\cdot r_{2,j}\\
=&\frac{1}{\sqrt{2\pi}\sigma_{2,j}}\cdot\prod_{i=5}^{m}\frac{1}{\sqrt{2\pi}\sigma_{i,j}}\cdot e^{\frac{-1}{2\sigma_{i,j}^2}(\phi_{i,j}-\mu_{i,j})^2}\cdot\frac{1}{\sqrt{2\pi}\sigma_4}\cdot e^{\frac{-1}{2\sigma_4^2}\left[\phi_4-(c_4\phi_{3,j}+d_4)\right]^2}\cdot\\
&e^{\frac{-1}{2\sigma_4^2}\left[\phi_{3,j}-(\bar{c}_{3,j}\phi_{1,j}+\bar{d}_{3,j})\right]^2}\cdot r_{2,j}
\end{aligned}
$$

Similarly, then we integrate $\phi_{3,j}$

$$
\begin{aligned}
& q(\phi_{m,j}, \phi_{m-1,j}, \ldots, \phi_4 | \phi_{1,j}) \\
= & \int q(\phi_{m,j}, \phi_{m-1,j}, \ldots, \phi_{3,j} | \phi_{1,j}) \, d\phi_{3,j} \\
= & \frac{r_{2,j}}{\sqrt{2\pi}\sigma_{2,j}} \cdot \prod_{i=5}^{m} \frac{1}{\sqrt{2\pi}\sigma_{i,j}} \cdot e^{\frac{-1}{2\sigma_{i,j}^2}(\phi_{i,j}-\mu_{i,j})^2} \cdot \int \frac{1}{\sqrt{2\pi}\sigma_4} \cdot e^{\frac{-1}{2\sigma_4^2}[\phi_4 - (c_4\phi_{3,j}+d_4)]^2} \cdot \\
& e^{\frac{-1}{2\sigma_4^2}\left[\phi_{3,j} - (\bar{c}_{3,j}\phi_{1,j}+\bar{d}_{3,j})\right]^2} \, d\phi_{2,j}.
\end{aligned}
$$

The formulation is similar to the previous one, so we can utilize the process above to integrate succesively, and we finally obtain

$$
q(\phi_{m,j}|\phi_{1,j}) = \frac{\prod_{i=2}^{m-1} r_{i,j}}{\sqrt{2\pi}\sigma_{2,j}} \cdot e^{\frac{-1}{2\sigma_{m,j}^2 \cdot (c_{m,j}^2+1)}\left[\phi_{m,j} - (\bar{c}_{m,j}\phi_{1,j}+\bar{d}_{m,j})\right]^2},
$$

which indicates

$$
p(\phi_{m,j}|\phi_{1,j}, D) \approx \frac{\prod_{i=2}^{m-1} r_{i,j}}{\sqrt{2\pi}\sigma_{2,j}} \cdot e^{\frac{-1}{2\sigma_{m,j}^2 \cdot (c_{m,j}^2+1)}\left[\phi_{m,j} - (\bar{c}_{m}\phi_{1,j}+\bar{d}_{m,j})\right]^2},
$$

where we have the iterative calculation by

$$
\begin{aligned}
r_{i,j} &= e^{\frac{d_{i+1,j}^2}{2\sigma_{i+1,j}^2}} \cdot \frac{1}{\sqrt{c_{i+1,j}^2+1}}, \\
\bar{c}_{i,j} &= c_{i,j}\bar{c}_{i-1,j}, \ (i \geq 3), \\
\bar{d}_{i,j} &= \bar{d}_{i-1,j}c_{i,j} + d_{i,j}, \ (i \geq 3).
\end{aligned}
$$

### A.3 Sampling from $p(\boldsymbol{\phi}_m|\boldsymbol{\phi}_M, D)$

To sample from the distribution $p(\boldsymbol{\phi}_m|\boldsymbol{\phi}_M, D)$, we first obtain a sample $\boldsymbol{\phi}_1$ from the prior distribution (*i.e.,* $\boldsymbol{\phi}_1 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$), then get $\boldsymbol{\phi}_m$ from a consecutive sampling process:

$$
\begin{aligned}
\boldsymbol{\phi}_{M-1} &\sim p(\boldsymbol{\phi}_{M-1}|\boldsymbol{\phi}_M, \boldsymbol{\phi}_1, D), \\
\boldsymbol{\phi}_{M-2} &\sim p(\boldsymbol{\phi}_{M-2}|\boldsymbol{\phi}_{M-1}, \boldsymbol{\phi}_1, D), \\
&\vdots \\
\boldsymbol{\phi}_{m} &\sim p(\boldsymbol{\phi}_{m}|\boldsymbol{\phi}_{m+1}, \boldsymbol{\phi}_1, D),
\end{aligned}
$$

because of the Markov property in our cascaded model. So our target is obtaining the distributions $p(\boldsymbol{\phi}_{i-1}|\boldsymbol{\phi}_i, \boldsymbol{\phi}_1, D)$. For a certain $p(\boldsymbol{\phi}_{i-1}|\boldsymbol{\phi}_i, \boldsymbol{\phi}_1, D)$, according to the Bayes rule, we have

$$
p(\boldsymbol{\phi}_{i-1}|\boldsymbol{\phi}_i, \boldsymbol{\phi}_1, D) = \frac{p(\boldsymbol{\phi}_i|\boldsymbol{\phi}_{i-1}, \boldsymbol{\phi}_1, D) \cdot p(\boldsymbol{\phi}_{i-1}|\boldsymbol{\phi}_1, D)}{p(\boldsymbol{\phi}_i|\boldsymbol{\phi}_1, D)}.
$$

Similarly with the last section, we use non-bold symbols to represent one dimension of the multi-dimension parameters, where they are able to transfer independently, and finally construct the eventual parameters by concatenating, that is,

$$
p(\boldsymbol{\phi}_{i-1}|\boldsymbol{\phi}_i, \boldsymbol{\phi}_1, D) = \prod_{j=1}^{d} p(\phi_{i,j}|\phi_{i-1,j}, \phi_{1,j}, D).
$$

So our target can be converted to calculate the probability $p(\phi_{i,j}|\phi_{i-1,j}, \phi_{1,j}, D)$ for all dimensions $\forall 1 \leq j \leq d$. According to the Markov property and the transportation probability, we have

$$
\begin{aligned}
p(\phi_{i,j}|\phi_{i-1,j}, \phi_{1,j}, D) &\approx q(\phi_{i,j}|\phi_{i-1,j}, \phi_{1,j}) \\
&= \frac{1}{\sqrt{2\pi}\sigma_{i,j}} \cdot e^{\frac{-1}{2\sigma_{i,j}^2} \cdot [\phi_{i,j} - (c_i\phi_{i-1,j}+d_{i,j})]^2}.
\end{aligned}
$$

According to the previous section, we have

$$p(\phi_{i,j}|\phi_{1,j},D) \approx q(\phi_{i,j}|\phi_{1,j}) = \frac{\prod_{i=2}^{i-1} r_{i,j}}{\sqrt{2\pi}\sigma_{2,j}} \cdot e^{\frac{-1}{2\sigma_{i,j}^2\cdot(c_{i,j}^2+1)}\left[\phi_{i,j}-(\bar{c}_i\phi_{1,j}+\bar{d}_i)\right]^2},$$

$$p(\phi_{i-1,j}|\phi_{1,j},D) \approx q(\phi_{i-1,j}|\phi_{1,j}) = \frac{\prod_{i=2}^{i-2} r_{i,j}}{\sqrt{2\pi}\sigma_{2,j}} \cdot e^{\frac{-1}{2\sigma_{i-1,j}^2\cdot(c_{i-1,j}^2+1)}\left[\phi_{i-1,j}-(\bar{c}_{i-1,j}\phi_{1,j}+\bar{d}_{i-1,j})\right]^2}.$$

Then we have

$$\begin{aligned}
p(\phi_{i-1,j}|\phi_{i,j},\phi_{1,j},D) &\approx \frac{q(\phi_{i,j}|\phi_{i-1,j},\phi_{1,j})\cdot q(\phi_{i-1,j}|\phi_{1,j})}{q(\phi_{i,j}|\phi_{1,j})} \\
=&\frac{1}{\sqrt{2\pi}\sigma_{i,j}\cdot r_{i-1}}\cdot e^{\frac{-1}{2\sigma_{i,j}^2}\cdot[\phi_{i,j}-(c_{i,j}\phi_{i-1,j}+d_{i,j})]^2}\cdot\frac{e^{\frac{[\phi_{i-1,j}-(\bar{c}_{i-1,j}\phi_{1,j}+\bar{d}_{i-1,j})]^2}{-2\sigma_{i-1,j}^2\cdot(c_{i-1,j}^2+1)}}}{e^{\frac{[\phi_{i,j}-(c_{i,j}\phi_{i-1,j}+d_{i,j})]^2}{-2\sigma_{i,j}^2\cdot(c_{i,j}^2+1)}}} \\
=&\sqrt{\frac{c_{i,j}+1}{2\pi\sigma_{i,j}^2}}\cdot e^{\frac{2\sigma_{i,j}^2}{d_{i,j}^2}}\cdot e^{\frac{[\phi_{i,j}-(c_{i,j}\phi_{i-1,j}+d_{i,j})]^2}{2\sigma_{i+1,j}^2}}\cdot e^{\frac{[\phi_{i,j}-(c_{i,j}\phi_{i-1,j}+d_{i,j})]^2}{-2\sigma_{i,j}^2}}\cdot e^{\frac{[\phi_{i-1,j}-(\bar{c}_{i-1,j}\phi_{1,j}+\bar{d}_{i-1,j})]^2}{-2\sigma_{i,j}^2}} \\
=&\sqrt{\frac{c_{i,j}+1}{2\pi\sigma_{i,j}^2}}\cdot e^{\frac{2\sigma_{i,j}^2}{d_{i,j}^2}+\frac{[\phi_{i,j}-(c_{i,j}\phi_{i-1,j}+d_{i,j})]^2}{2\sigma_{i+1,j}^2}}. \\
&e^{\frac{1}{-2\sigma_{i,j}^2}\cdot\underbrace{\left\{[\phi_{i,j}-(c_{i,j}\phi_{i-1,j}+d_{i,j})]^2+\left[\phi_{i-1,j}-(\bar{c}_{i-1,j}\phi_{1,j}+\bar{d}_{i-1,j})\right]^2\right\}}_{C}}
\end{aligned}$$

Then we calculate the part $C$ as

$$\begin{aligned}
C &= [\phi_{i,j}-(c_{i,j}\phi_{i-1,j}+d_{i,j})]^2+\left[\phi_{i-1,j}-(\bar{c}_{i-1,j}\phi_{1,j}+\bar{d}_{i-1,j})\right]^2 \\
&= \phi_{i,j}^2+(c_{i,j}\phi_{i-1,j}+d_{i,j})^2-2(c_{i,j}\phi_{i-1,j}+d_{i,j})\phi_{i,j} \\
&+ \phi_{i-1,j}^2+(\bar{w}_{i-1}\phi_{1,j}+\bar{d}_{i-1,j})^2-2\phi_{i-1,j}(\bar{c}_{i-1,j}\phi+\bar{d}_{i-1,j}) \\
&= \phi_{i,j}^2+c_{i,j}^2\phi_{i-1,j}^2+d_{i,j}^2+2d_{i,j}c_{i,j}\phi_{i-1,j}-2c_{i,j}\phi_{i-1,j}\phi_{i,j}-2d_{i,j}\phi_{i,j}+\phi_{i-1,j}^2 \\
&+ \bar{c}_{i-1,j}^2\phi_{1,j}^2+\bar{d}_{i-1,j}^2+2\bar{c}_{i-1,j}\bar{d}_{i-1,j}\phi_{1,j}-2\bar{c}_{i-1,j}\phi_{1,j}\phi_{i-1,j}-2\bar{d}_{i-1,j}\phi_{i-1,j} \\
&= (\bar{w}_{i-1}^2+1)\cdot\left[\phi_{i-1,j}-\frac{c_{i,j}\phi_{i,j}+\bar{c}_{i-1,j}\phi_{1,j}+\bar{d}_{i-1,j}-d_{i,j}c_{i,j}}{c_{i,j}^2+1}\right]^2+B,
\end{aligned}$$

where $B$ does not include $\phi_i$, which indicates

$$p(\phi_{i-1,j}|\phi_{i,j},\phi_{1,j},D) \sim \mathcal{N}\left(\frac{c_{i,j}\phi_{i,j}+\bar{c}_{i-1,j}\phi_{1,j}+\bar{d}_{i-1,j}-d_{i,j}w_{i-2}}{c_{i,j}^2+1},\frac{\sigma_{i,j}^2}{c_{i,j}^2+1}\right).$$

### A.4 Calculation of $p(\boldsymbol{x}|\boldsymbol{e},\boldsymbol{\phi}_m)$

In this section, we will show how to calculate the graph probability $p(\boldsymbol{x}|\boldsymbol{e},\boldsymbol{\phi}_m)$. Remember the graph parameters $\boldsymbol{\phi}_m = [\boldsymbol{\theta}_m;\mathbf{S}_m;\mathbf{T}_m]$, so we have

$$\begin{aligned}
p(\boldsymbol{x}|\boldsymbol{e},\boldsymbol{\phi}_m) &= \int_{\mathbf{E}} p(\boldsymbol{x}|\boldsymbol{e},\boldsymbol{\theta}_m,\mathbf{E})\cdot p(\mathbf{E}|\boldsymbol{e},\boldsymbol{S}_m,\boldsymbol{T}_m)\,d\mathbf{E} \\
&= \mathbb{E}_{\mathbf{E}\sim p(\mathbf{E}|\boldsymbol{S}_m,\boldsymbol{T}_m)}\left[p(\boldsymbol{x}|\boldsymbol{e},\boldsymbol{\theta}_m,\mathbf{E})\right].
\end{aligned}$$

According to Monte Carlo sampling, we have

$$p(\boldsymbol{x}|\boldsymbol{e},\boldsymbol{\phi}_m) = \frac{1}{K}\cdot\sum_{l=1}^{K} p(\boldsymbol{x}|\boldsymbol{e},\boldsymbol{\theta}_m,\mathbf{E}_l),$$

where $\mathbf{E}_l[i,j] \sim \text{Bernoulli}(\sigma(\mathbf{S}_m^T[i]\cdot\mathbf{T}_m[j]))$. In order to conduct intervention process, we change the $j$th column of $\mathbf{E}_l$ to zeros, and represent it with $\tilde{\mathbf{E}}_l$. Moreover, we replace the $j$th element of $\boldsymbol{x}$

with $v$, and get the result $\tilde{\boldsymbol{x}}$. We change the $j$th element of $\boldsymbol{\epsilon}_m$ with zero, and get the result $\tilde{\boldsymbol{\epsilon}}_m$. Then according the definition of causal graphs, we have

$$p(\boldsymbol{x}|\boldsymbol{e}, \boldsymbol{\phi}_m) = \frac{1}{K}\sum_{l=1}^{K} \mathcal{N}(\boldsymbol{x}; \boldsymbol{f}(\tilde{\boldsymbol{x}}; \tilde{\mathbf{E}}_l, \boldsymbol{\gamma}_m), \tilde{\boldsymbol{\epsilon}}_m),$$

where $\boldsymbol{f}$ is the causal function that depends on the parameter $\boldsymbol{\gamma}_m$.

# B Bayesian Optimization for Determining $(j^*, v^*, m^*)$

We intend to find the best tuple for acquisition, that is,

$$(j^*, v^*, m^*) = \arg\max_{(j,v,m)} f(j, v, m).$$

We define the best interventional value $v$ under interventional node $j$ and fidelity $m$ as

$$\begin{aligned} v^*(j, m) &= \arg\max_{v} f(j, v, m) \\ &= \arg\max_{v} f_{j,m}(v). \end{aligned}$$

where $f_{j,m}(v)$ is rewritten from $f(j, v, m)$ under given $j, m$. Therefore, our task is calculating $v^*(j, m)$ for $\forall j \in [d], m \in [M]$ with Bayesian optimization [1]. We utilize a Gaussian Process (GP) [2] to model surrogate function distributions for each $v^*(j, m)$. We denote $f \sim \mathcal{GP}(\mathbf{0}, \mathcal{K}(v_i, v_j))$, and $\mathcal{K}(v_i, v_j)$ is the kernel of GP. We sequentially find $v_t$ and calculate $f_{j,m}(v_t)$ to direct the process. According to GP, the previous $t$ functions and the $t+1$ function are multivariate Gaussian distribution,

$$\begin{bmatrix} \mathbf{F}_{1:t} \\ f_{t+1} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_t & \boldsymbol{k}_{t+1} \\ \boldsymbol{k}_{t+1}^T & \mathcal{K}(v_{t+1}, v_{t+1}) \end{bmatrix}\right),$$

where we define

$$\begin{aligned} \mathbf{F}_{1:t} &= [f_1, f_2, \dots, f_t], \\ \boldsymbol{k}_{t+1} &= [\mathcal{K}(v_{t+1}, v_1), \mathcal{K}(v_{t+1}, v_2), \dots, \mathcal{K}(v_{t+1}, v_{t+1})]^T, \\ \mathbf{K}_t &= \begin{bmatrix} \mathcal{K}(v_1, v_1) & \cdots & \mathcal{K}(v_t, v_1) \\ \vdots & \ddots & \vdots \\ \mathcal{K}(v_t, v_1) & \cdots & \mathcal{K}(v_t, v_t) \end{bmatrix}. \end{aligned} \tag{1}$$

Given previous $t$ steps, we have the posterior probability is

$$p(f_{t+1}|\{(v_i, f_{j,m}(v_i))\}_{i=1}^{t}, v_{t+1}) = \mathcal{N}(\mu_t(v_{t+1}), \sigma_t^2(v_{t+1})),$$

with the non-parametric means and variances

$$\mu_t(v_{t+1}) = \boldsymbol{k}_{t+1}^T(\mathbf{K} + \mathbf{I})^{-1}\mathbf{F}_{1:t}, \tag{2}$$

$$\sigma_t^2(v_{t+1}) = \mathcal{K}(v_{v+1}, v_{t+1}) - \boldsymbol{k}_{t+1}^T(\mathbf{K} + \mathbf{I})^{-1}\boldsymbol{k}_{t+1}. \tag{3}$$

We acquire the next $v_{t+1}$ with GP-UCB [3] function

$$\begin{aligned} a_{t+1}(v) &= \mu_t(v) + \beta_{ac} \cdot \sqrt{\sigma_t^2(v)}, \\ v_{t+1} &= \arg\max_{v} a_{t+1}(v). \end{aligned}$$

where $\beta_{ac}$ is a hyper-parameter. Suppose the maximum of steps is $T$, the final output of function $v^*(j, m)$ is

$$v^*(j, m) = \arg\max_{v} \mu_T(v).$$

Then we choose the best interventional node $j$ and fidelity $m$ by their best values under $\mathcal{O}(d \cdot M)$

$$\begin{aligned} j^*, m^* &= \arg\max_{j,m} v^*(j, m), \\ v^* &= v^*(j^*, m^*). \end{aligned}$$

# C Detailed Training Process of ELBO

## C.1 Derivation Process of ELBO

Because we use the distribution $q(\boldsymbol{\phi}_m)$ to approximate the distribution $p(\boldsymbol{\phi}_m)$, then we intend to minimize the distance between these two distributions optimize the parameters of $q(\boldsymbol{\phi}_m)$, where we utilize KL divergence to measure the distance, that is,

$$\Psi^* = \arg\min_{\Psi} \mathrm{KL}[q(\boldsymbol{\Phi}||p(\boldsymbol{\Phi}|D)].$$

According to the variational inference, we have

$$\begin{aligned}
&\mathrm{KL}\left[q(\boldsymbol{\Phi})||p(\boldsymbol{\Phi}|D)\right] \\
&= \int q(\boldsymbol{\Phi}) \log \frac{q(\boldsymbol{\Phi})}{p(\boldsymbol{\Phi}|D)}\, d\boldsymbol{\Phi} \\
&= \int q(\boldsymbol{\Phi}) \log q(\boldsymbol{\Phi})\, d\boldsymbol{\Phi} - \int q(\boldsymbol{\Phi}) \log p(\boldsymbol{\Phi}|D)\, d\boldsymbol{\Phi} \\
&= \mathbb{E}_{\boldsymbol{\Phi}\sim q(\boldsymbol{\Phi})}\left[\log q(\boldsymbol{\Phi})\right] - \int q(\boldsymbol{\Phi}) \log \frac{p(\boldsymbol{\Phi}, D)}{p(D)}\, d\boldsymbol{\Phi} \\
&= \mathbb{E}_{\boldsymbol{\Phi}\sim q(\boldsymbol{\Phi})}\left[\log q(\boldsymbol{\Phi})\right] - \int q(\boldsymbol{\Phi}) \log p(\boldsymbol{\Phi}, D)\, d\boldsymbol{\Phi} + \int q(\boldsymbol{\Phi}) \log p(D)\, d\boldsymbol{\Phi} \\
&= \mathbb{E}_{\boldsymbol{\Phi}\sim q(\boldsymbol{\Phi})}\left[\log q(\boldsymbol{\Phi})\right] - \mathbb{E}_{\boldsymbol{\Phi}\sim q(\boldsymbol{\Phi})}\left[\log p(\boldsymbol{\Phi}, D)\right] + \int q(\boldsymbol{\Phi}) \log p(D)\, d\boldsymbol{\Phi} \\
&= \underbrace{\mathbb{E}_{\boldsymbol{\Phi}\sim q(\boldsymbol{\Phi})}\left[\log q(\boldsymbol{\Phi})\right] - \mathbb{E}_{\boldsymbol{\Phi}\sim q(\boldsymbol{\Phi})}\left[\log p(\boldsymbol{\Phi}, D)\right]}_{-\mathrm{ELBO}} + \log p(D).
\end{aligned}$$

Because $\log p(D)$ is not related to $\Psi$, minimizing $\mathrm{KL}\left[q(\boldsymbol{\Phi})||p(\boldsymbol{\Phi}|D)\right]$ is equivalent to maximizing the ELBO part, and we have

$$\begin{aligned}
\mathrm{ELBO} &= \mathbb{E}_{\boldsymbol{\Phi}\sim q(\boldsymbol{\Phi})}\left[\log p(\boldsymbol{\Phi}, D)\right] - \mathbb{E}_{\boldsymbol{\Phi}\sim q(\boldsymbol{\Phi})}\left[\log q(\boldsymbol{\Phi})\right] \\
&= \mathbb{E}_{\boldsymbol{\Phi}\sim q(\boldsymbol{\Phi})}\left[\log p(D|\boldsymbol{\Phi})\right] + \mathbb{E}_{\boldsymbol{\Phi}\sim q(\boldsymbol{\Phi})}\left[\log p(\boldsymbol{\Phi})\right] - \mathbb{E}_{\boldsymbol{\Phi}\sim q(\boldsymbol{\Phi})}\left[\log q(\boldsymbol{\Phi})\right] \\
&= \mathbb{E}_{\boldsymbol{\Phi}\sim q(\boldsymbol{\Phi})}\left[\log p(D|\boldsymbol{\Phi}) - \log q(\boldsymbol{\Phi}) + \log p(\boldsymbol{\Phi})\right]
\end{aligned}$$

Above all, we can conclude that

$$\Psi^* = \arg\min_{\Psi} \mathrm{KL}[q(\boldsymbol{\Phi}||p(\boldsymbol{\Phi}|D)]$$

is equivalent to maximize evidence lower bound

$$\begin{aligned}
\Psi^* &= \arg\max_{\Psi} \mathrm{ELBO} \\
&= \arg\max_{\Psi} \mathbb{E}_{\boldsymbol{\Phi}\sim q(\boldsymbol{\Phi})}\left[\log p(D|\boldsymbol{\Phi}) - \log q(\boldsymbol{\Phi}) + \log p(\boldsymbol{\Phi})\right].
\end{aligned}$$

## C.2 Estimation of ELBO

We represent the equation of ELBO as

$$\begin{aligned}
\mathrm{ELBO} &= \mathbb{E}_{\boldsymbol{\Phi}\sim q(\boldsymbol{\Phi})}\left[\log p(D|\boldsymbol{\Phi}) - \log q(\boldsymbol{\Phi}) + \log p(\boldsymbol{\Phi})\right] \\
&= \underbrace{\mathbb{E}_{\boldsymbol{\Phi}\sim q(\boldsymbol{\Phi})}\left[\log p(D|\boldsymbol{\Phi})\right]}_{A} - \underbrace{\mathbb{E}_{\boldsymbol{\Phi}\sim q(\boldsymbol{\Phi})}\left[\log q(\boldsymbol{\Phi}) - \log p(\boldsymbol{\Phi})\right]}_{B}.
\end{aligned}$$

For the part $A$, we have

$$A = \mathbb{E}_{\boldsymbol{\Phi}\sim q(\boldsymbol{\Phi})}\left[\log \prod_{i=1}^{N} p(\boldsymbol{x}^{(i)}|j^{(i)}, v^{(i)}, m^{(i)}, \boldsymbol{\Phi})\right],$$

where $N$ is the current number of samples in buffer. Then we have

$$
\begin{aligned}
A =& \mathbb{E}_{\mathbf{\Phi} \sim q(\mathbf{\Phi})} \left[ \log \prod_{i=1}^{N} p(\boldsymbol{x}^{(i)} | j^{(i)}, v^{(i)}, m^{(i)}, \mathbf{\Phi}) \right] \\
=& \mathbb{E}_{\mathbf{\Phi} \sim q(\mathbf{\Phi})} \left[ \sum_{i=1}^{N} \log p(\boldsymbol{x}^{(i)} | j^{(i)}, v^{(i)}, m^{(i)}, \mathbf{\Phi}) \right] \\
=& \sum_{i=1}^{N} \mathbb{E}_{\mathbf{\Phi} \sim q(\mathbf{\Phi})} \left[ \log p(\boldsymbol{x}^{(i)} | j^{(i)}, v^{(i)}, m^{(i)}, \mathbf{\Phi}) \right] \\
=& \sum_{i=1}^{N} \mathbb{E}_{\boldsymbol{\phi}_{m^{(i)}} \sim q(\boldsymbol{\phi}_{m^{(i)}})} \left[ \log p(\boldsymbol{x}^{(i)} | j^{(i)}, v^{(i)}, m^{(i)}, \boldsymbol{\phi}_{m^{(i)}}) \right].
\end{aligned}
$$

Using Monte Carlo sampling [4], we can calculate the expectation by $N_S$ samples for each point.

$$
A = \sum_{i=1}^{N} \sum_{j=1}^{N_S} \log p(\boldsymbol{x}^{(i)} | j^{(i)}, v^{(i)}, m^{(i)}, \boldsymbol{\phi}_{m^{(i)}}^{(j)}),
$$

where we sample $\boldsymbol{\phi}_{m^{(i)}}^{(j)} \sim q(\boldsymbol{\phi}_{m^{(i)}})$ with size $N_S$.

Then we denote the distribution $q(\mathbf{\Phi}) = \mathcal{N}(\tilde{\boldsymbol{\mu}}_{all}, \tilde{\mathbf{\Sigma}}_{all})$, and similarly, we have $p(\mathbf{\Phi}) = \prod_{m=1}^{M} e^{-\beta \cdot f(\mathbf{S}_m, \mathbf{T}_m)} \cdot \mathcal{N}(\boldsymbol{\mu}_{all}, \mathbf{\Sigma}_{all})$. Both the parameter $\tilde{\boldsymbol{\mu}}_{all}, \tilde{\mathbf{\Sigma}}_{all}$ can be represented by the parameters in $\Psi$, while $\boldsymbol{\mu}_{all}$ and $\mathbf{\Sigma}_{all}$ are constant. Then we calculate part $B$

$$
\begin{aligned}
B =& \mathbb{E}_{\mathbf{\Phi} \sim q(\mathbf{\Phi})} \left[ \log q(\mathbf{\Phi}) - \log p(\mathbf{\Phi}) \right] \\
=& \int_{\mathbf{\Phi}} \mathcal{N}(\tilde{\boldsymbol{\mu}}_{all}, \tilde{\mathbf{\Sigma}}_{all}) \log \frac{\mathcal{N}(\tilde{\boldsymbol{\mu}}_{all}, \tilde{\mathbf{\Sigma}}_{all})}{\prod_{m=1}^{M} e^{-\beta \cdot f(\mathbf{S}_m, \mathbf{T}_m)} \cdot \mathcal{N}(\boldsymbol{\mu}_{all}, \mathbf{\Sigma}_{all})} \, d\mathbf{\Phi} \\
=& \int_{\mathbf{\Phi}} \mathcal{N}(\tilde{\boldsymbol{\mu}}_{all}, \tilde{\mathbf{\Sigma}}_{all}) \log \frac{\mathcal{N}(\tilde{\boldsymbol{\mu}}_{all}, \tilde{\mathbf{\Sigma}}_{all})}{\mathcal{N}(\boldsymbol{\mu}_{all}, \mathbf{\Sigma}_{all})} \, d\mathbf{\Phi} + \int_{\mathbf{\Phi}} \mathcal{N}(\tilde{\boldsymbol{\mu}}_{all}, \tilde{\mathbf{\Sigma}}_{all}) \log \frac{1}{\prod_{m=1}^{M} e^{-\beta \cdot f(\mathbf{S}_m, \mathbf{T}_m)}} \, d\mathbf{\Phi} \\
=& \underbrace{\mathrm{KL}[\mathcal{N}(\tilde{\boldsymbol{\mu}}_{all}, \tilde{\mathbf{\Sigma}}_{all}) || \mathcal{N}(\boldsymbol{\mu}_{all}, \mathbf{\Sigma}_{all})]}_{C} + \underbrace{\int_{\mathbf{\Phi}} \mathcal{N}(\tilde{\boldsymbol{\mu}}_{all}, \tilde{\mathbf{\Sigma}}_{all}) \log \prod_{m=1}^{M} e^{\beta \cdot f(\mathbf{S}_m, \mathbf{T}_m)} \, d\mathbf{\Phi}}_{D}.
\end{aligned}
$$

According to KL divergence of Gaussian distribution, we can calculate $C$ in a close-form.

$$
\begin{aligned}
C =& \mathrm{KL}[\mathcal{N}(\tilde{\boldsymbol{\mu}}_{all}, \tilde{\mathbf{\Sigma}}_{all}) || \mathcal{N}(\boldsymbol{\mu}_{all}, \mathbf{\Sigma}_{all})] \\
=& \frac{1}{2} \left[ \log \frac{||\mathbf{\Sigma}_{all}||}{||\tilde{\mathbf{\Sigma}}_{all}||} - d + \mathrm{tr}(\mathbf{\Sigma}_{all}^{-1} \tilde{\mathbf{\Sigma}}_{all}) + (\tilde{\boldsymbol{\mu}}_{all} - \boldsymbol{\mu}_{all})^T \mathbf{\Sigma}_{all}^{-1} (\tilde{\boldsymbol{\mu}}_{all} - \boldsymbol{\mu}_{all}) \right].
\end{aligned}
$$

Then we calculate $D$ by the following steps:

$$
\begin{aligned}
D =& \int_{\mathbf{\Phi}} \mathcal{N}(\boldsymbol{\mu}_{all}, \mathbf{\Sigma}_{all}) \log \prod_{m=1}^{M} e^{\beta \cdot f(\mathbf{S}_m, \mathbf{T}_m)} \, d\mathbf{\Phi} \\
=& \int_{\mathbf{\Phi}} \mathcal{N}(\boldsymbol{\mu}_{all}, \mathbf{\Sigma}_{all}) \sum_{m=1}^{M} \log e^{\beta \cdot f(\mathbf{S}_m, \mathbf{T}_m)} \, d\mathbf{\Phi} \\
=& \cdot \int_{\mathbf{\Phi}} \mathcal{N}(\boldsymbol{\mu}_{all}, \mathbf{\Sigma}_{all}) \sum_{m=1}^{M} \beta \cdot f(\mathbf{S}_m, \mathbf{T}_m) \, d\mathbf{\Phi} \\
=& \beta \cdot \mathbb{E}_{\mathbf{\Phi} \sim \mathcal{N}(\boldsymbol{\mu}_{all}, \mathbf{\Sigma}_{all})} \left[ \sum_{m=1}^{M} f(\mathbf{S}_m, \mathbf{T}_m) \right].
\end{aligned}
$$

Using Monte Carlo sampling, we can calculate the expectation by $N_D$ samples for each point.

$$
\begin{aligned}
D &= \beta \cdot \sum_{i=1}^{N_D} \sum_{m=1}^{M} f(\mathbf{S}_m^{(i)}, \mathbf{T}_m^{(i)}). \\
&= \beta \cdot \sum_{i=1}^{N_D} \sum_{m=1}^{M} \mathbb{E}_{p(\mathbf{E}|\mathbf{S}_m^{(i)}, \mathbf{T}_m^{(i)})} \left[\lambda_1 \cdot \left[\text{tr}\left(e^{\mathbf{E}}\right) - d\right] + \lambda_2 \cdot ||\mathbf{E}||\right],
\end{aligned}
$$

where we samples $\mathbf{\Phi}^{(i)} \sim \mathcal{N}(\boldsymbol{\mu}_{all}, \mathbf{\Sigma}_{all})$ with size $N_D$. Using Monte Carlo sampling again, we can calculate the expectation by $N_E$ samples.

$$
D = \beta \cdot \sum_{i=1}^{N_D} \sum_{m=1}^{M} \sum_{j=1}^{N_E} \left[\lambda_1 \cdot \left[\text{tr}\left(e^{\mathbf{E}}\right) - d\right] + \lambda_2 \cdot ||\mathbf{E}||\right],
$$

where we samples $\mathbf{E}^{(j)} \sim p(\mathbf{E}|\mathbf{S}_m^{(i)}, \mathbf{T}_m^{(i)})$ with size $N_E$.

Finally, we obtain the estimation

$$
\begin{aligned}
\text{ELBO} = &\sum_{i=1}^{N} \sum_{j=1}^{N_S} \log p(x^{(i)}|j^{(i)}, v^{(i)}, m^{(i)}, \boldsymbol{\phi}_{m^{(i)}}^{(j)}) \\
&- \frac{1}{2}\left[\log \frac{||\mathbf{\Sigma}_{all}||}{||\tilde{\mathbf{\Sigma}}_{all}||} - d + \text{tr}(\mathbf{\Sigma}_{all}^{-1}\tilde{\mathbf{\Sigma}}_{all}) + (\tilde{\boldsymbol{\mu}}_{all} - \boldsymbol{\mu}_{all})^T \mathbf{\Sigma}_{all}^{-1}(\tilde{\boldsymbol{\mu}}_{all} - \boldsymbol{\mu}_{all})\right] \\
&- \beta \cdot \sum_{i=1}^{N_D} \sum_{m=1}^{M} \sum_{j=1}^{N_E} \left[\lambda_1 \cdot \left[\text{tr}\left(e^{\mathbf{E}}\right) - d\right] + \lambda_2 \cdot ||\mathbf{E}||\right].
\end{aligned}
$$

### C.3 Gaussian Reparameterization Trick

In the last section, we derive the objection function for optimizing the model parameters, where we can use methods of the gradient decent to solve it. However, a significant problem rises due to the sampling process, because the gradient of model parameters can not pass backward from the naive sampling process(*i.e.,* untraceable). Therefore, we use Gaussian reparameterization trick to make the Gaussian sampling process traceable.

In specific, we will demonstrate the traceable calculation of $\boldsymbol{\phi}$ by Gaussian reparameterization trick. In order to sample $\boldsymbol{\phi} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$, we first sample $\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ instead, and then obtain $\boldsymbol{\phi} = \boldsymbol{\mu} + \boldsymbol{\delta} \odot \mathbf{\Sigma}$. Therefore, the gradient can be traced from $\boldsymbol{\phi}$ to $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$. In specific, both $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$ can be represented with the function of learnable parameter $\Psi$.

### C.4 Gumbel-softmax Reparameterization Trick

Besides of the Gaussian sampling process, the Bernoulli sampling in our equation is not traceable either, so we utilize Gumbel-softmax reparameterization trick to make it traceable.

We demonstrate the traceable calculation of $\mathbf{E} \sim p(\mathbf{E}|\mathbf{S}, \mathbf{T})$ by Gumbel-max reparameterization trick. According to Gumbel-max [5], we have

$$
\text{Bernoulli}(p) \iff \mathbf{1}\left[G_1 + \log p > G_0 + \log(1-p)\right], \quad G_0, G_1 \sim \text{Gumbel}(0, 1).
$$

Instead of using unit step function, we utilize sigmoid function

$$
\sigma(G_1 + \log p > G_0 + \log(1-p)).
$$

Therefore, we have

$$
\mathbf{E}_{i,j} = \sigma(\mathbf{L}_{i,j} + \mathbf{S}_i^T \cdot \mathbf{T}_j),
$$

where $\mathbf{L}_{i,j} \sim L(0, 1)$. Therefore, we sample $\mathbf{L}_{i,j} \sim L(0, 1)$ instead, where $L(0, 1)$ is logistic distribution, and calculate $\mathbf{E}_{i,j} = \sigma(\mathbf{L}_{i,j} + \mathbf{S}_i^T \cdot \mathbf{T}_j)$ to trace gradients. Specifically, both $\boldsymbol{S}_i$ and $\boldsymbol{T}_i$ can be represented with the function of learnable parameter $\Psi$.

### C.5 Optimization of ELBO

With the estimation and reparameterization trick, we are able to conduct gradient descent methods to optimize our parameters with the objection function

$$\Psi^* = \arg\max_{\Psi} \text{ELBO}.$$

The format of stochastic gradient descent (SGD) is

$$\Psi \leftarrow \Psi + \gamma \cdot \frac{\partial \text{ELBO}}{\partial \Psi},$$

where $\gamma$ is the learning rate.

## D Training Process of Constraint based ELBO

We intend to optimize our parameter with

$$\begin{aligned} \Psi^* =& \arg\max_{\Psi} \mathbb{E}_{\mathbf{\Phi}\sim q(\mathbf{\Phi})}\left[\log p(D|\mathbf{\Phi}) - \log q(\mathbf{\Phi}) + \log p(\mathbf{\Phi})\right], \\ &\text{s.t.} \sum_{\{\boldsymbol{e}_s, \boldsymbol{e}_t\}} I(\boldsymbol{x}_s; \boldsymbol{x}_t | \boldsymbol{\phi}_M, \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) \leq \epsilon. \end{aligned}$$

However, the objection has a constraint, which is hard to optimize with gradient descent methods. So we utilize Lagrange multiplier [6] to convert it to a constraint-free method:

$$\Psi^* = \arg\max_{\Psi} \mathbb{E}_{\mathbf{\Phi}\sim q(\mathbf{\Phi})}\left[\log p(D|\mathbf{\Phi}) - \log q(\mathbf{\Phi}) + \log p(\mathbf{\Phi})\right] + \lambda \cdot \sum_{\{\boldsymbol{e}_s, \boldsymbol{e}_t\}} I(\boldsymbol{x}_s; \boldsymbol{x}_t | \boldsymbol{\phi}_M, \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D),$$

where $\lambda$ is the Lagrange multiplier. Then, we intend to calculate the constraint part.

First of all, we have

$$\begin{aligned} & I(\boldsymbol{x}_s; \boldsymbol{x}_t | \boldsymbol{\phi}_M, \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) \\ =& H(\boldsymbol{x}_s | \boldsymbol{\phi}_M, \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) + H(\boldsymbol{x}_t | \boldsymbol{\phi}_M, \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) - H(\boldsymbol{x}_s, \boldsymbol{x}_t | \boldsymbol{\phi}_M, \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) \\ =& H(\boldsymbol{x}_s | \boldsymbol{\phi}_M, \boldsymbol{e}_s, D) + H(\boldsymbol{x}_t | \boldsymbol{\phi}_M, \boldsymbol{e}_t, D) - H(\boldsymbol{x}_s, \boldsymbol{x}_t | \boldsymbol{\phi}_M, \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D), \end{aligned}$$

For the term $H(\boldsymbol{x}_s | \boldsymbol{\phi}_M, \boldsymbol{e}_s, D)$, we have

$$\begin{aligned} H(\boldsymbol{x}_s | \boldsymbol{\phi}_M, \boldsymbol{e}_s, D) &= -\int p(\boldsymbol{x}_s | \boldsymbol{\phi}_M, \boldsymbol{e}_s, D) \log p(\boldsymbol{x}_s | \boldsymbol{\phi}_M, \boldsymbol{e}_s, D)\, d\boldsymbol{x}_s \\ &= -\mathbb{E}_{p(\boldsymbol{x}_s | \boldsymbol{\phi}_M, \boldsymbol{e}_s, D)}\left[\log p(\boldsymbol{x}_s | \boldsymbol{\phi}_M, \boldsymbol{e}_s, D)\right]. \end{aligned}$$

We use Monte Carlo sampling to estimate $H(\boldsymbol{x}_s | \boldsymbol{\phi}_M, \boldsymbol{e}_s, D)$, and we have

$$H(\boldsymbol{x}_s | \boldsymbol{\phi}_M, \boldsymbol{e}_s, D) \approx \frac{1}{K_1 \cdot K_2} \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \log p(\boldsymbol{x}^{(k_1,k_2)} | \boldsymbol{e}^s, \boldsymbol{\phi}_M),$$

where we sample graphs $\boldsymbol{\phi}_m^{k_1} \sim q(\boldsymbol{\phi}_m | \boldsymbol{e}^s, \boldsymbol{\phi}_M)$, and obtain samples $\boldsymbol{x}^{(k_1,k_2)} \sim p(\boldsymbol{x} | \boldsymbol{e}^s, \boldsymbol{\phi}_m^{k_1})$. Similarly, we can calculate

$$H(\boldsymbol{x}_t | \boldsymbol{\phi}_M, \boldsymbol{e}_t, D) \approx \frac{1}{K_1 \cdot K_2} \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \log p(\boldsymbol{x}^{(k_1,k_2)} | \boldsymbol{e}^t, \boldsymbol{\phi}_M),$$

where we sample graphs $\boldsymbol{\phi}_m^{k_1} \sim q(\boldsymbol{\phi}_m | \boldsymbol{e}^t, \boldsymbol{\phi}_M)$, and obtain samples $\boldsymbol{x}^{(k_1,k_2)} \sim p(\boldsymbol{x} | \boldsymbol{e}^t, \boldsymbol{\phi}_m^{k_1})$.

And we have

$$H(\boldsymbol{x}_s, \boldsymbol{x}_t | \boldsymbol{\phi}_M, \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) \approx \frac{1}{K_1 \cdot K_2 \cdot K_3} \sum_{k_1=1}^{K_1} \sum_{k_2^1=1}^{K_2} \sum_{k_2^2=1}^{K_2} \log p(\boldsymbol{x}^{(k_1,k_2^1)}, \boldsymbol{x}^{(k_1,k_2^2)} | \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, \boldsymbol{\phi}_M),$$

where we sample graphs $\boldsymbol{\phi}_m^{k_1} \sim q(\boldsymbol{\phi}_m|\{\boldsymbol{e}_s, \boldsymbol{e}_t\}, \boldsymbol{\phi}_M)$, obtain samples $\boldsymbol{x}^{(k_1,k_2^1)} \sim p(\boldsymbol{x}|\boldsymbol{e}^s, \boldsymbol{\phi}_m^{k_1})$, and obtain samples $\boldsymbol{x}^{(k_1,k_2^2)} \sim p(\boldsymbol{x}|\boldsymbol{e}^t, \boldsymbol{\phi}_m^{k_1})$.

Therefore, we add constraint on the original loss function to obtained the estimation of constraint based ELBO, that is,

$$
\begin{aligned}
\text{ELBO} = &\sum_{i=1}^{N}\sum_{j=1}^{N_S} \log p(x^{(i)}|j^{(i)}, v^{(i)}, m^{(i)}, \boldsymbol{\phi}_{m^{(i)}}^{(j)}) \\
&- \frac{1}{2}\left[\log\frac{||\boldsymbol{\Sigma}_{all}||}{||\tilde{\boldsymbol{\Sigma}}_{all}||} - d + \text{tr}(\boldsymbol{\Sigma}_{all}^{-1}\tilde{\boldsymbol{\Sigma}}_{all}) + (\tilde{\boldsymbol{\mu}}_{all} - \boldsymbol{\mu}_{all})^T \boldsymbol{\Sigma}_{all}^{-1}(\tilde{\boldsymbol{\mu}}_{all} - \boldsymbol{\mu}_{all})\right] \\
&- \beta \cdot \sum_{i=1}^{N_D}\sum_{m=1}^{M}\sum_{j=1}^{N_E}\left[\lambda_1 \cdot \left[\text{tr}\left(e^{\mathbf{E}}\right) - d\right] + \lambda_2 \cdot ||\mathbf{E}||\right] + \lambda \cdot \left[I(\boldsymbol{x}_s; \boldsymbol{x}_t|\boldsymbol{\phi}_M, \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D)\right].
\end{aligned}
$$

# E Proof of Theory 3

*Proof.* To begin with, we introduce two anchor variables $\boldsymbol{x}, \boldsymbol{e}$, indicating existing samples and experiments in the system, which are independent with the following experiments. Since $\boldsymbol{x}_s, \boldsymbol{x}_t$ are $\epsilon$-independent given $\boldsymbol{\phi}_M, \{\boldsymbol{e}_s, \boldsymbol{e}_t\}$ and $D$, we have:

$$
\begin{aligned}
&I(\boldsymbol{x}_s; \boldsymbol{x}_t|\boldsymbol{\phi}_M, \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) = I(\boldsymbol{x}_s; \boldsymbol{x}_t|\boldsymbol{\phi}_M, \boldsymbol{x}, \boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) \leq \epsilon \\
\Leftrightarrow\ &H(\boldsymbol{x}_s|\boldsymbol{\phi}_M, \boldsymbol{x}, \boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) + H(\boldsymbol{x}_t|\boldsymbol{\phi}_M, \boldsymbol{x}, \boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) \\
&- H(\boldsymbol{x}_s, \boldsymbol{x}_t|\boldsymbol{\phi}_M, \boldsymbol{x}, \boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) \leq \epsilon,
\end{aligned}
$$

Since

$$
\begin{aligned}
I(\boldsymbol{x}_s; \boldsymbol{\phi}_M|\boldsymbol{x}, \boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) =& H(\boldsymbol{x}_s|\boldsymbol{x}, \boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) - H(\boldsymbol{x}_s|\boldsymbol{\phi}_M, \boldsymbol{x}, \boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) \\
I(\boldsymbol{x}_t; \boldsymbol{\phi}_M|\boldsymbol{x}, \boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) =& H(\boldsymbol{x}_t|\boldsymbol{x}, \boldsymbol{e} \cup \{\boldsymbol{e}_t, \boldsymbol{e}_t\}, D) - H(\boldsymbol{x}_t|\boldsymbol{\phi}_M, \boldsymbol{x}, \boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D)
\end{aligned}
$$

We have:

$$
\begin{aligned}
&I(\boldsymbol{x}_s; \boldsymbol{\phi}_M|\boldsymbol{x}, \boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) + I(\boldsymbol{x}_t; \boldsymbol{\phi}_M|\boldsymbol{x}, \boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) \\
=& H(\boldsymbol{x}_s|\boldsymbol{x}, \boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) + H(\boldsymbol{x}_t|\boldsymbol{x}, \boldsymbol{e} \cup \{\boldsymbol{e}_t, \boldsymbol{e}_t\}, D) \\
-& H(\boldsymbol{x}_s|\boldsymbol{\phi}_M, \boldsymbol{x}, \boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) - H(\boldsymbol{x}_t|\boldsymbol{\phi}_M, \boldsymbol{x}, \boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) \\
\geq& H(\boldsymbol{x}_s, \boldsymbol{x}_t|\boldsymbol{x}, \boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) - H(\boldsymbol{x}_s, \boldsymbol{x}_t|\boldsymbol{\phi}_M, \boldsymbol{x}, \boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) - \epsilon \\
=& I(\boldsymbol{x}_s, \boldsymbol{x}_t; \boldsymbol{\phi}_M|\boldsymbol{x}, \boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) - \epsilon.
\end{aligned}
$$

According to the basic mutual information property $I(A, B; C) - I(B; C) = I(A; C|B)$, we have:

$$
\begin{aligned}
&I(\boldsymbol{x} \cup \boldsymbol{x}_s; \boldsymbol{\phi}_M|\boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) - I(\boldsymbol{x}; \boldsymbol{\phi}_M|\boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) \\
+& I(\boldsymbol{x} \cup \boldsymbol{x}_t; \boldsymbol{\phi}_M|\boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) - I(\boldsymbol{x}; \boldsymbol{\phi}_M|\boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) \\
\geq& I(\boldsymbol{x} \cup \{\boldsymbol{x}_t, \boldsymbol{x}_s\}; \boldsymbol{\phi}_M|\boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) - I(\boldsymbol{x}; \boldsymbol{\phi}_M|\boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) - \epsilon.
\end{aligned}
$$

Thus, we have:

$$
\begin{aligned}
&I(\boldsymbol{x} \cup \boldsymbol{x}_s; \boldsymbol{\phi}_M|\boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) + I(\boldsymbol{x} \cup \boldsymbol{x}_t; \boldsymbol{\phi}_M|\boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) \\
\geq& I(\boldsymbol{x} \cup \{\boldsymbol{x}_t, \boldsymbol{x}_s\}; \boldsymbol{\phi}_M|\boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) + I(\boldsymbol{x}; \boldsymbol{\phi}_M|\boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) - \epsilon.
\end{aligned}
$$

Since different experiments are independent, we have:

$$
\begin{aligned}
&I(\boldsymbol{x} \cup \boldsymbol{x}_s; \boldsymbol{\phi}_M|\boldsymbol{e} \cup \{\boldsymbol{e}_s\}, D) + I(\boldsymbol{x} \cup \boldsymbol{x}_t; \boldsymbol{\phi}_M|\boldsymbol{e} \cup \{\boldsymbol{e}_t\}, D) \\
\geq& I(\boldsymbol{x} \cup \{\boldsymbol{x}_t, \boldsymbol{x}_s\}; \boldsymbol{\phi}_M|\boldsymbol{e} \cup \{\boldsymbol{e}_s, \boldsymbol{e}_t\}, D) + I(\boldsymbol{x}; \boldsymbol{\phi}_M|\boldsymbol{e}, D) - \epsilon.
\end{aligned}
$$

Thus, $I(\cdot; \boldsymbol{\phi}_M|\cdot, D)$ is $\epsilon$-submodular. □

# F Proof of Theory 4

For clear presentation, we denote $g(\{\boldsymbol{e}_i\}_{i=1}^n) = I(\{\boldsymbol{x}_i\}_{i=1}^n; \boldsymbol{\phi}_M|\{\boldsymbol{e}_i\}_{i=1}^n, D)$, then we need to solve the following problem:

$$
\arg\max_{\{\boldsymbol{e}_i\}_{i=1}^n} g(\{\boldsymbol{e}_i\}_{i=1}^n), \tag{4}
$$

Suppose $S^* = \{\boldsymbol{e}_i^*\}_{i=1}^n$ is the optimal solution for objective (4), and the results of the greedy method is $S = \{\boldsymbol{e}_i\}_{i=1}^n$, where the experiments are sequentially determined from $\boldsymbol{e}_1$ to $\boldsymbol{e}_n$. We denote $S_{1:j} = \{\boldsymbol{e}_i\}_{i=1}^j$, and $\Delta(\boldsymbol{e}|S_{1:j}) = g(S_{1:j} \cup \boldsymbol{e}) - g(S_{1:j})$, according to the greedy method, we have:

$$\boldsymbol{e}_{j+1} = \arg\max_{\boldsymbol{e}} \frac{\Delta(\boldsymbol{e}|S_{1:j})}{\lambda_{\boldsymbol{e}}},$$

where $\lambda_{\boldsymbol{e}}$ is the cost of experiment $\boldsymbol{e}$.

Based on all the above notations, we have:

$$\begin{aligned}
g(S^*) &\leq g(S^* \cup S_{1:j}) \\
&= g(S_{1:j}) + g(S_{1:j} \cup \boldsymbol{e}_1^*) - g(S_{1:j}) \\
&+ g(S_{1:j} \cup \boldsymbol{e}_1^* \cup \boldsymbol{e}_2^*) - g(S_{1:j} \cup \boldsymbol{e}_1^*) \\
&+ \dots \\
&+ g(S_{1:j} \cup \{\boldsymbol{e}_1^*, ..., \boldsymbol{e}_n^*\}) - g(X_{1:i} \cup \{\boldsymbol{e}_1^*, ..., \boldsymbol{e}_{n-1}^*\}) \\
&= g(S_{1:j}) + \sum_{k=1}^{n} \left[ g(S_{1:j} \cup \{\boldsymbol{e}_1^*, ..., \boldsymbol{e}_k^*\}) - g(X_{1:i} \cup \{\boldsymbol{e}_1^*, ..., \boldsymbol{e}_{k-1}^*\}) \right] \\
&\leq g(S_{1:j}) + \sum_{k=1}^{n} \left[ g(S_{1:j} \cup \{\boldsymbol{e}_k^*\}) - g(S_{1:j}) + \epsilon \right] \\
&= g(S_{1:j}) + \sum_{k=1}^{n} \left[ \Delta(\{\boldsymbol{e}_k^*\}|S_{1:j}) + \epsilon \right],
\end{aligned}$$

where the first inequality holds because of the non-decreasing property, and the second inequality holds because of the $\epsilon$-submodular property.

Since $\boldsymbol{e}_{j+1} = \arg\max_{\boldsymbol{e}} \frac{\Delta(\boldsymbol{e}|S_{1:j})}{\lambda_{\boldsymbol{e}}}$, we have $\frac{\Delta(\boldsymbol{e}|S_{1:j})}{\lambda_{\boldsymbol{e}}} \leq \frac{\Delta(\boldsymbol{e}_{j+1}|S_{1:j})}{\lambda_{\boldsymbol{e}_{j+1}}}$ for any $\boldsymbol{e}$, thus $\Delta(\boldsymbol{e}|S_{1:j}) \leq \frac{\lambda_{\boldsymbol{e}}}{\lambda_{\boldsymbol{e}_{j+1}}} \Delta(\boldsymbol{e}_{j+1}|S_{1:j}) \leq B_\lambda \Delta(\boldsymbol{e}_{j+1}|S_{1:j})$. By bringing this result into the above equation, we have:

$$\begin{aligned}
g(S^*) &\leq g(S_{1:j}) + \sum_{k=1}^{n} \left[ \Delta(\{\boldsymbol{e}_k^*\}|S_{1:j}) + \epsilon \right] \\
&\leq g(S_{1:j}) + \sum_{k=1}^{n} \left[ B_\lambda \Delta(\boldsymbol{e}_{j+1}|S_{1:j}) + \epsilon \right] \\
&= g(S_{1:j}) + n B_\lambda \Delta(\boldsymbol{e}_{j+1}|S_{1:j}) + n\epsilon
\end{aligned}$$

Let $T_j = g(S^*) - g(S_{1:j})$, we have:

$$T_j - T_{j+1} = g(S_{1:j+1}) - g(S_{1:j}) = \Delta(\boldsymbol{e}_{j+1}|S_{1:j}) \geq \frac{T_j - n\epsilon}{nB_\lambda}$$

Then

$$\begin{aligned}
T_n &\leq (1 - \frac{1}{nB_\lambda}) T_{n-1} + \frac{\epsilon}{B_\lambda} \leq [(1 - \frac{1}{nB_\lambda})]^2 T_{n-2} + (1 - \frac{1}{nB_\lambda}) \frac{\epsilon}{B_\lambda} + \frac{\epsilon}{B_\lambda} \\
&\leq ... \leq [(1 - \frac{1}{nB_\lambda})]^n T_0 + [(1 - \frac{1}{nB_\lambda})]^{n-1} \frac{\epsilon}{B_\lambda} + ... + \frac{\epsilon}{B_\lambda}
\end{aligned}$$

Let $B = [(1 - \frac{1}{nB_\lambda})]^{n-1} \frac{\epsilon}{B_\lambda} + ... + \frac{\epsilon}{B_\lambda} = \frac{\epsilon}{B_\lambda} \sum_{i=1}^{n} [(1 - \frac{1}{nB_\lambda})]^{i-1}$, and considering that $[(1 - \frac{1}{nB_\lambda})]^n = e^{-\frac{1}{B_\lambda}}$, we have:

$$g(S^*) - g(S_{1:n}) \leq e^{-\frac{1}{B_\lambda}} g(S^*) + B$$

Thus, we have $g(S_{1:n}) \geq (1 - e^{-\frac{1}{B_\lambda}}) g(S^*) - B$.

# G Algorithm

The algorithm for Licence method for single-target interventiion scenario is shown in Algorithm 1. Moreover, the algorithm for Licence method for multi-target interventiion scenario is shown in Algorithm 2.

**Algorithm 1:** Algorithm of Licence for Single-target Intervention Scenario

**Input:** Variable set $X_V$, number of oracles $M$, cost of oracles $\mathbf{\Lambda}$, observational data $D^O$, total budget $C$, and learning rate $\eta$.
**Output:** Causal graph $\boldsymbol{\phi}_M$.

Initialize the model parameter $\Psi$ .
Optimize $\Psi$ with the training process of ELBO under $D^O$.
Initialize $D^I = \emptyset$.
**while** *Budget* $C$ *does not run out* **do**
    Initialize $j^*, m^*, v^*$ and let $\zeta^* = -\infty$.
    **for** $(j, m)$ *in* $\{1, 2, \ldots, d\} \times \{1, 2, \ldots, M\}$ **do**
        Calculate $v^*(j, m)$ with BO.
        **if** $f(j, v^*(j, m), m) > \zeta^*$ **then**
            Update $j^* \leftarrow j, m^* \leftarrow m$ and $v^* \leftarrow v^*(j, m)$.
            Update $\zeta^* \leftarrow f(j, v^*(j, m), m)$.
        **end**
    **end**
    Subtract the budget with $C \leftarrow C - \lambda_{m^*}$.
    Acquire $(j^*, v^*, m^*)$ towards the true causal graph to obtain $\boldsymbol{x}^* \sim p_m(X_V | do(X_j = v))$.
    Update $D^I \leftarrow D^I \cup \{\boldsymbol{x}^*\}$.
    Optimize $\Psi$ with training process of ELBO under $D^O \cup D^I$.
**end**
Sample $\boldsymbol{\phi}_M$ from $p(\boldsymbol{\phi}_M | D)$
**return** *Causal graph* $\boldsymbol{\phi}_M$.

# H More Experiments

## H.1 Experimental Settings

### H.1.1 Datasets

The details of our experimental datasets are presented as follows:

• **Erdős-Rényi (ER)** [7] graph is a random graph introduced by Paul Erdős and Alfréd Rényi. For ER graph, a graph with $n$ vertices is generated by connecting each pair of vertices with a probability $p$.

• **Scale-Free (SF)** [8] graph is a type of random graph that has a degree distribution following power law. A small number of vertices in SF graph own a large number of edges, while the vast majority of vertices have relatively few edges.

• **DREAM** [9] is the abbreviation for Dialogue for Reverse Engineering Assessments and Methods, which can estimate the reverse quality that causal discovery methods perform. Specifically, we use a biological graph generator GeneNetWeaver for our experiments, which is a real-word public dataset.

### H.1.2 Baselines

The details of experimental baselines are demonstrated as follows. We utilize DiBS [10] as our basic graph representation component. For acquisition methods, we use AIT and CBED and obtain the query tuples of node and value.

• **AIT** [11] is an active learning method that utilize f-score to select intervention queries.

• **CBED** [12] is based on the calculation of mutual information (MI), which intend to select intervention queries with maximal MI scores after obtaining new samples under current queries.

For the multi-target intervention scenario, we extend above methods with greedy strategy, which can promise an lower bound for approximation with submodular property. For choosing the fidelities to query, we use two circumstances, *i.e.,* REAL and RANDOM.

**Algorithm 2:** Algorithm of Licence for Multi-target Intervention Scenario

**Input:** Variable set $X_V$, number of oracles $M$, cost of oracles $\mathbf{\Lambda}$, observational data $D^O$, total multi-target experiment step $T$, total budget $C$, and learning rate $\eta$.
**Output:** Causal graph $\boldsymbol{\phi}_M$.

Initialize the model parameter $\Psi$ .
Optimize $\Psi$ with training process of constraint based ELBO under $D^O$.
Initialize $B^I = \emptyset$
**for** $t$ *in* $1, 2, \ldots, T$ **do**
  **while** *Budget C does not run out* **do**
    Initialize $j^*, m^*, v^*$ and let $\zeta^* = -\infty$.
    **for** $(j, m)$ *in* $\{1, 2, \ldots, d\} \times \{1, 2, \ldots, M\}$ **do**
      Calculate $v^*(j, m)$ with BO.
      **if** $f(j, v^*(j, m), m) > \zeta^*$ **then**
        Update $j^* \leftarrow j, m^* \leftarrow m$ and $v^* \leftarrow v^*(j, m)$.
        Update $\zeta^* \leftarrow f(j, v^*(j, m), m)$.
      **end**
    **end**
    Subtract the budget with $C \leftarrow C - \lambda_{m^*}$.
    Update $B^I \leftarrow B^I \cup \{(j^*, v^*, m^*)\}$.
  **end**
  Acquire $B^I$ towards the true causal graph to obtain $\{\boldsymbol{x}^* \sim p_m(X_V|do(X_j = v))\}_{(j,v,m)\in B^I}$.
  Update $D^I \leftarrow D^I \cup \{\boldsymbol{x}^*\}_{(j,v,m)\in B^I}$.
  Optimize $\Psi$ with training process of constraint based ELBO under $D^O \cup D^I$.
**end**
Sample $\boldsymbol{\phi}_M$ from $p(\boldsymbol{\phi}_M|D)$
**return** *Causal graph* $\boldsymbol{\phi}_M$.

• **REAL** fidelity means the model always choose the highest fidelity to conduct experiments. This strategy is aligned with classic causal discovery under active learning paradigm without multi-fidelity settings, which can just choose the most accurate samples to conduct discovery process.

• **RANDOM** fidelity means the model choose different fidelities randomly with uniform probability.

#### H.1.3 Metrics

The details of experimental metrics are demonstrated as follows. We utilize SHD and AUPRC to reflect the topological structure discovering performance, and design MSE to reflex the predicting performance of functional relations.

• **SHD** [13] is the abbreviation for Structural Hamming Distance, and it estimate the topological structure by counting the number of different edges on adjacency matrix. We calculate the expectation of SHD under multiple graph samplings.

• **AUPRC** [14] is the area under precision-recall curve, where we consider entities on the adjacency matrix as binary classification problem. The AUPRC is also under the expectation for multiple graph sampling.

• **MSE** is designed for estimating the performance of grasping functional relations. We obtain several samples from the true causal graph, and let our model and the true causal function to conduct forward process respectively, then calculate the MSE between the two results. We calculate MSE by sampling graphs for multiple times.

### H.2 Details of Configurations and Computation

The details of the configurations of device and platform are demonstrate in Table 1(left). We will show the details of the time cost on computation. We measure the time cost on the generation of each intervention per fidelity for all models, and the results are shown in Figure 1(right). We find that our

Table 1: The left table demonstrate the details of the configuration of device and platform. The right table shows the details of time cost on computation.

| Name | Details |
|---|---|
| CPU | Intel Xeon Platinum 8350C 2.60GHz |
| GPU | RTX A5000 (24GB) |
| Memory | 42GB RAM |
| Python | Version 3.8 |
| Java | Version 1.8.0 (Necessary for DREAM) |

| Model | Time (secs) |
|---|---|
| AIT-REAL | 7.686 |
| AIT-RANDOM | 7.451 |
| CBED-REAL | 7.998 |
| CBED-RANDOM | 7.989 |
| Licence | 8.320 |

Table 2: The details of experimental settings.

| Name | Explanation | Value |
|---|---|---|
| budget | The total budget for interventional experiments, (*i.e.,* C). | 10/20/30/40/50 |
| oracle number | The number of oracles, (*i.e.,* $M$) | 3 |
| oracle cost | The cost for each oracle, (*i.e.,* $\mathbf{\Lambda}$) | 2, 8, 32 |
| oracle noise | The extra additive noise for each oracle. | 0.04, 0.02, 0.00 |
| observation number | The number of observational samples. | 1000 |
| expect edge number | The number of expect edges. | 2 |
| additive noise | The value of additive noise during data generations. | 0.01 |

method cost a little more than the baselines, which is probably due to the more complex sampling process in our model.

We also show the details of experimental settings for our overall experiments in Table 2. We carefully tune the hyper-parameters for baselines and our model, and the final values can be obtained in the configuration file in our codes.

### H.3 Experiments on DREAM Dataset

We conduct experiments on a real-world biological dataset, called DREAM. Note that, DREAM does not support the calculation of MSE, because of the lack of interface in this real-world dataset. We use two sub-datasets *Ecoli* and *Yeast* as our true causal graphs. The results are shown in Figure 1. We find that our model outperforms that other baselines on both *Ecoli* and *Yeast*, and both single-target and multi-target intervention scenario.

### H.4 Experiments on More Nodes

In this section, we conduct further experiments on datasets with more nodes. We extend the number of nodes from 10 to 20, and experiment on the ER graph. The results are shown in Figure 3. We find that our model is still effective on the scenario of more nodes, and is better than baselines.

## I Potentially Negative Social Impact

Causal discovery focuses on understanding causal relationships between variables. While causal discovery has the potential to bring about positive social impacts, it is important to consider both the positive and negative implications of its applications. In this response, I will focus on the negative impact of causal discovery.

• **Reductionism and Oversimplification.** Causal discovery techniques often aim to identify simple cause-and-effect relationships. However, complex social phenomena often involve a multitude of interconnected factors, making it difficult to capture the full complexity of the system. Relying solely on causal discovery may lead to oversimplification and reductionism, neglecting the nuanced interactions between variables.

• **Ethical Concerns.** Causal discovery can involve analyzing sensitive data, such as personal information or medical records. If not handled carefully, the use of this data can raise significant ethical concerns related to privacy, consent, and potential discrimination. Improper handling of data could lead to violations of privacy and unfair treatment of individuals or groups.

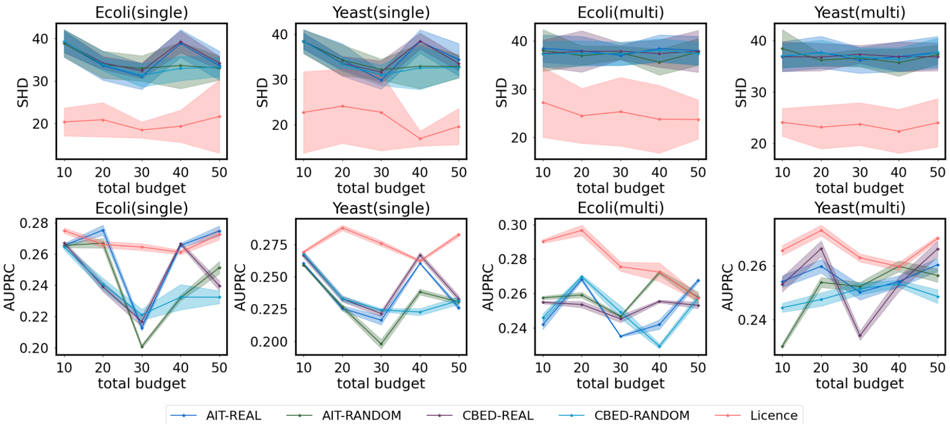


Figure 1: The performance among models on DREAM datasets with different datasets and budgets. Lower SHD, MSE indicate better performances. We conduct each experiment for ten times, and report the average performances and error bars.

Table 3: SHD results of 20 nodes graphs on different budgets. Lower SHD indicates better performances. We conduct each experiment for ten times, and report average performances and error bars.

| Model | Budget(10) | Budget(20) | Budget(30) | Budget(40) | Budget(50) |
|---|---|---|---|---|---|
| AIT-REAL | 63.36±4.89 | 64.36±5.18 | 64.53±6.83 | 63.28±4.86 | 64.35±5.19 |
| AIT-RANDOM | 63.62±4.61 | 62.16±5.75 | 64.60±5.23 | 66.87±6.47 | 63.53±5.27 |
| DiBS-REAL | 63.58±6.35 | 61.50±7.69 | 63.50±6.86 | 63.56±6.34 | 61.45±7.69 |
| DiBS-RANDOM | 63.68±6.77 | 65.07±6.41 | 63.91±7.14 | 63.99±4.46 | 63.86±3.00 |
| Licence | 49.67±11.64 | 49.61±8.08 | 55.68±8.63 | 51.34±11.24 | 51.36±9.11 |

• **Overreliance on Correlation.** Causal discovery often relies on identifying statistical correlations between variables. However, correlation does not imply causation, and there is a risk of mistakenly inferring causal relationships based solely on correlation. Overreliance on such methods can lead to erroneous conclusions, leading to misguided decision-making and ineffective interventions.

• **Social Bias and Inequality.** Causal discovery relies on the data used for analysis, which can reflect existing biases and inequalities present in society. If the data used is biased, the causal relationships discovered may perpetuate or exacerbate existing social inequalities. Causal discovery methods need to be sensitive to potential biases and strive for fairness and inclusivity in both data collection and analysis.

In conclusion, while causal discovery holds promise in understanding complex systems, it is crucial to consider its potential negative impacts. Oversimplification, ethical concerns, overreliance on correlation, and social bias are all factors that need to be addressed to ensure responsible and beneficial applications of causal discovery. It is essential to approach this field with caution and incorporate broader societal considerations to mitigate the negative impacts and harness its potential for positive social change.

## References


[1] Harold J Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. 1964.

[2] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures*, pages 63–71. Springer, 2004.

[3] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[4] Reuven Y Rubinstein and Dirk P Kroese. *Simulation and the Monte Carlo method*. John Wiley & Sons, 2016.

[5] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[6] Dimitri P Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

[7] Paul Erdős, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

[8] Lun Li, David Alderson, John C Doyle, and Walter Willinger. Towards a theory of scale-free graphs: Definition, properties, and implications. *Internet Mathematics*, 2(4):431–523, 2005.

[9] Thomas Schaffter, Daniel Marbach, and Dario Floreano. Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, 27(16): 2263–2270, 2011.

[10] Lars Lorch, Jonas Rothfuss, Bernhard Schölkopf, and Andreas Krause. Dibs: Differentiable bayesian structure learning. *Advances in Neural Information Processing Systems*, 34:24111–24123, 2021.

[11] Yashas Annadani, Jonas Rothfuss, Alexandre Lacoste, Nino Scherrer, Anirudh Goyal, Yoshua Bengio, and Stefan Bauer. Variational causal networks: Approximate bayesian inference over causal structures. *arXiv preprint arXiv:2106.07635*, 2021.

[12] Panagiotis Tigas, Yashas Annadani, Andrew Jesson, Bernhard Schölkopf, Yarin Gal, and Stefan Bauer. Interventions, where and how? experimental design for causal models at scale. In *Advances in Neural Information Processing Systems*.

[13] Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65:31–78, 2006.

[14] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.