Supplementary Materials for "`cyclinbayes`: Bayesian Causal Discovery with Linear Non-Gaussian Directed Acyclic and Cyclic Graphical Models"

Robert Lee [1], Raymond K. W. Wong[1], and Yang Ni[2*]

1 Department of Statistics, Texas A&M University, College Station, TX 77840, USA.

2 Department of Statistics and Data Sciences, The University of Texas at Austin, Austin, TX 78705, USA.

∗ To whom correspondence should be addressed: yang.ni@austin.utexas.edu

1 Background

Discovering causal relationships among variables is a fundamental task across many scientific disciplines, and causal discovery in high-dimensional settings has received growing attention in recent years. Several methods leverage the non-Gaussian error assumption, including LiNGAM [Shimizu et al., 2006] and DirectLiNGAM [Shimizu et al., 2011], which are implemented in R packages `pcalg` [Kalisch et al., 2012] and `rlingam` [Kikuchi, 2025]. While these

methods represent substantial methodological progress beyond constraint- or score-based methods, most operate within a purely frequentist framework and return only a single estimated structure without providing measures of structural or parameter uncertainty. Moreover, nearly all available implementations are restricted to directed acyclic graphs (DAGs) and lack support for modeling feedback or reciprocal causal relationships. Recent work has begun to accommodate reciprocal structure in more specialized settings; for example, `MR.RGM` implements Bayesian reciprocal graphical models for multivariate bidirectional Mendelian randomization networks [Sarkar and Ni, 2025]. However, general-purpose Bayesian LiNGAM software for learning cyclic graphs with full posterior uncertainty quantification-beyond Markov equivalence classes—remains largely unavailable in R. Although algorithms such as CCD for learning Markov equivalence classes of directed cyclic graphs (DCGs) exist in both Python and R, Bayesian implementations for LiNGAM or DCG learning beyond Markov equivalence classes are scarce, and available code is often unmaintained and not reliably runnable in modern software environments.

To address limitations, we introduce `cyclinbayes`, an open source R package for flexible and scalable Bayesian causal discovery in both DAGs and DCGs. By sampling from the full posterior distribution over graph structures and causal effects, the package enables principled uncertainty quantification, including posterior edge inclusion probabilities, credible intervals for model parameters such as direct causal effects, and posterior probabilities of user-specified network motifs. To obtain a simple representative graph, we adopt a decision-theoretic framework that selects the structure minimizing the posterior weighted average distance to all sampled graphs under metrics such as the Structural Hamming Distance (SHD), Structural Intervention Distance (SID), and any user-specified graph distance. Implemented in Rcpp, `cyclinbayes` leverages optimized C++ routines to efficiently handle large-scale, high-dimensional datasets, offering a unified and computationally robust toolkit for Bayesian causal discovery with and without feedback. The package is available on GitHub

at `https://github.com/roblee01/cyclinbayes`.

2 Model Specification

Consider a random vector

$$\vec{\mathbf{Y}}_i = (Y_i^{(1)}, \ldots, Y_i^{(q)}, \ldots, Y_i^{(N)})^T \quad \text{for } i = 1, \ldots, p.$$

Let $Y_i^{(q)}$ be the qth realization of ith random variable or feature for $i = 1, ..., p$ and $q = 1, \ldots, N$.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, \ldots, p\}$ denotes the set of nodes corresponding to the p random variables, and $\mathcal{E}$ denotes the set of directed edges representing direct causal relationships among these variables. For DAGs, the graph is acyclic, meaning that one cannot start from a node and, by following the directed edges, return to the same node. DCGs, on the other hand, allow the presence of directed cycles so that a path may begin and end at the same node. Let $pa(i)$ be the set of parents (direct causes) for node $i \in \mathcal{V}$. For a node $i \in \mathcal{V}$, let $pa(i) \subseteq \{1, \ldots, p\} \setminus \{i\}$ denote its parent set.

We consider the following linear structural causal model with mixture of Gaussians errors,

$$Y_i^{(q)} = \sum_{j \in pa(i)} B_{ij} Y_j^{(q)} + \epsilon_i^{(q)}, \tag{1}$$

where $j \in pa(i)$ means that there is a directed edge from $j \to i$, B_{ij} represents the direct causal effect from a variable $Y_j^{(q)}$ to another $Y_i^{(q)}$, and $\epsilon_i^{(q)} \sim \sum_{k=1}^{M} \pi_{ik} N(\mu_{ik}, \tau_{ik})$. We denote $\vec{\mu}_i = (\mu_{i1}, \ldots, \mu_{iM})$ and $\vec{\tau}_i = (\tau_{i1}, \ldots, \tau_{iM})$. We can write (1) in a matrix form,

$$\vec{\mathbf{Y}}^{(q)} = B\vec{\mathbf{Y}}^{(q)} + \vec{\boldsymbol{\epsilon}}^{(q)}, \tag{2}$$

where $\vec{\mathbf{Y}}^{(q)} = (Y_1^{(q)}, \ldots, Y_p^{(q)})$ and $\vec{\epsilon}^{(q)} = (\epsilon_1^{(q)}, \ldots, \epsilon_p^{(q)})^T$ are p-dimensional vectors and $B = (B_{ij})$ is an unknown $p \times p$ direct causal effect matrix subject to algorithmic constraints, including $B_{ii} = 0$ for all $i \in \{1, \ldots, p\}$.

3 Bayesian Causal Discovery Algorithms

We present Bayesian inference procedures for both DAGs and DCGs within a hierarchical framework. The goal is to estimate a sparse adjacency matrix E encoding the causal structure.

3.1 Prior Specification

Let $E = (E_{ij})_{i,j=1}^{p}$ denote the adjacency matrix, where $E_{ij} = 1$ indicates a directed edge $j \to i$. We enforce $B_{ij} = 0$ whenever $E_{ij} = 0$, so sparsity in E directly induces sparsity in B.

Adjacency matrix. Edges are included independently with probability γ:

$$E_{ij} \mid \gamma \sim \text{Bernoulli}(\gamma), \qquad \gamma \sim \text{Beta}(a_\gamma, b_\gamma).$$

In `BayesDAG`, the matrix E is constrained to be acyclic; `BayesDCG` permits directed cycles except for self-loops.

Causal effect matrix. We use a conditional spike-and-slab prior:

$$B_{ij} \mid E_{ij}, \gamma_1 \sim (1 - E_{ij})\, \delta_0 + E_{ij}\, N(0, \gamma_1), \qquad \gamma_1 \sim \text{IG}(a_{\gamma_1}, b_{\gamma_1}).$$

Mixture error distribution. For Gaussian mixture errors, we assign conjugate priors:

$$\pi_i \sim \text{Dirichlet}(\alpha, \ldots, \alpha), \quad \mu_{ik} \sim N(a_\mu, b_\mu), \quad \tau_{ik} \sim \text{IG}(a_\tau, b_\tau).$$

For ease of sampling, we augment the Gaussian mixture with categorical variables $Z_i^{(q)}$ such that

$$Z_i^{(q)} \mid \pi_i \sim \text{Categorical}(\pi_{i1}, \ldots, \pi_{iM}),$$

$$\epsilon_i^{(q)} | Z_i^{(q)} = k \sim N(\mu_{ik}, \tau_{ik}),$$

for $i = 1, \ldots, p,\ q = 1, \ldots, N$.

3.2 DAG Structure Estimation

For DAGs, the likelihood is given by

$$p(\vec{Y}_1, \ldots, \vec{Y}_p \mid B, \boldsymbol{\mu}, \boldsymbol{\tau}, \mathbf{Z}) = \prod_{i=1}^{p} \prod_{k=1}^{M} \prod_{q=1}^{N} \left[N\left(\mu_{ik} + \sum_{j \in \mathrm{pa}(i)} B_{ij} Y_j^{(q)},\, \tau_{ik} \right) \right]^{I(Z_i^{(q)}=k)}.$$

We integrate out B to obtain the marginal posterior of E, which has a closed form (Appendix A). However, direct sampling is infeasible since the space of E is enormous. Instead, we use a Metropolis–Hastings (MH) scheme with two steps for each pair (i, j):

1. (Edge addition or deletion) Propose $E_{ij} = 1$ if currently $E_{ij} = 0$ (similarly, propose $E_{ij} = 0$ if $E_{ij} = 1$). Reject the new graph if it is not acyclic. If it is acyclic, accept/reject with MH-ratio:
$$r = \frac{p(\vec{Y}_1, \ldots, \vec{Y}_p \mid E_{ij} = 1, E_{-(ij)})}{p(\vec{Y}_1, \ldots, \vec{Y}_p \mid E_{ij} = 0, E_{-(ij)})}.$$
where $E_{-(ij)} := \{E_{kl} : (k, l) \neq (i, j)\}$.

2. (Edge reversal) For existing edges, propose to reverse its direction. Reject the new graph if it is not acyclic. Otherwise, accept/reject it with MH ratio:
$$\tilde{r} = \frac{p(\vec{Y}_1, \ldots, \vec{Y}_p | E_{ij} = 1, E_{ji} = 0, E_{-\{(i,j),(j,i)\}})}{p(\vec{Y}_1, \ldots, \vec{Y}_p | E_{ji} = 0, E_{ij} = 1, E_{-\{(i,j),(j,i)\}})}.$$

The marginalized forms of numerator and denominator of r and $\tilde{r}$ are derived in Appendix A.9.

To address label switching in the mixture model [McLachlan and Peel, 2004], we order the component means: $\mu_{i1} \leq \mu_{i2} \leq \cdots \leq \mu_{iM}$.

3.3 DCG Structure Estimation

For DCGs, the lack of a recursive factorization prevents analytical marginalization of B. We therefore jointly sample (E, B). To ensure stability [Lacerda et al., 2012], we reject any proposal for which the spectral radius $\rho(B) = \max_{\lambda \in \mathrm{eig}(B)} |\lambda| \geq 1$. For edge addition,

we propose $E_{ij} = 1$ with $B_{ij} \sim N(0, \sigma^2_{\text{add}})$ and calculate the MH ratio (Appendix B). For existing edges, we use random-walk updates $B'_{ij} \sim N(B_{ij}, \sigma^2_{\text{rw}})$.

4 Algorithms

The MCMC algorithms for DAGs and DCGs are provided in Algorithms 1 and 2, respectively. Full conditional distributions are provided in Appendices A and B. One key feature of Algorithm 1 is the collapsed Gibbs step for E: we marginalize over B to obtain a closed-form marginal likelihood ratio (Appendix A.9). Simulated annealing in the first half of iterations helps escape local modes. Unlike Algorithm 1, in Algorithm 2, we sample E and B jointly since marginalization is not available for cyclic graphs, and we reject proposals if the spectral radius $\rho(B) \geq 1$.

With the MCMC algorithm, we obtain posterior samples of $\{\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(m)}\}$, which provide a distribution over graph structures. A common approach thresholds edge-wise posterior inclusion probabilities, but this treats edges independently and may produce graphs violating model constraints (e.g., acyclicity or stability).

Instead, we cast graph selection as a Bayes decision problem. Let $d(\cdot, \cdot)$ be a graph distance. The Bayes optimal graph minimizes posterior expected loss:

$$\hat{\mathcal{G}} = \arg\min_{a \in \mathcal{D}} \sum_{g \in \mathcal{D}} d(a, g)\, \pi(g \mid \vec{Y}),$$

where $\mathcal{D}$ is the collection of DAGs under consideration. We approximate both the minimization and marginalization by posterior samples. Let $\{\mathcal{G}_*^{(1)}, \ldots, \mathcal{G}_*^{(v)}\}$ be the unique sampled graphs with frequencies $w_1, \ldots, w_v$. We select the graph minimizing the weighted sum of distances:

$$\hat{\mathcal{G}} = \arg\min_{l \in \{1,\ldots,v\}} \sum_{u=1}^{v} w_u\, d(\mathcal{G}_*^{(l)}, \mathcal{G}_*^{(u)}).$$

The distance metrics that our package supports are: (1) **SHD** (Structural Hamming Distance): number of edge additions, deletions, or reversals; (2) **SID** (Structural Interven-

tion Distance, DAGs only): number of pairs with differing interventional distributions; (3) **Custom**: any user-supplied graph distances.

Algorithm 1 BayesDAG: Bayesian Sparse Collapsed LiNGAM for DAGs

Initialize $\gamma, \gamma_1, E, B, \{\pi_i, \boldsymbol{\mu}_i, \boldsymbol{\tau}_i, \mathbf{Z}_i\}_{i=1}^{p}$
for $m = 1, \ldots, M_{\text{iter}}$ **do**
    Sample $\gamma \mid E$ from Beta full conditional
    for $i = 1, \ldots, p$ **do**
        Sample $\tau_{ik} \mid \boldsymbol{\epsilon}_i, \mathbf{Z}_i$ from Inverse-Gamma, $k = 1, \ldots, M$
    end for
    for each pair (i, j) with $i \neq j$ **do**
        // Edge addition/deletion step
        Compute $r = [\mathbf{Y} \mid E_{ij} = 1, E_{-(ij)}]/[\mathbf{Y} \mid E_{ij} = 0, E_{-(ij)}]$ via marginalization
        Apply simulated annealing: $r_m = r \cdot (c_2 r)^{\eta(m)}$ for $m < M_{\text{iter}}/2$
        Accept $E_{ij} = 1$ with probability $\min(1, r_m)$ if acyclic
        // Edge reversal step (if edge exists)
        Compute $\tilde{r}$ comparing $(E_{ij}, E_{ji}) = (1, 0)$ vs. $(0, 1)$
        Accept direction with probability based on $\tilde{r}_m$
    end for
    for $i = 1, \ldots, p$ **do**
        Sample $\mu_{ik} \mid \boldsymbol{\epsilon}_i, \mathbf{Z}_i, \tau_{ik}$ from Normal, $k = 1, \ldots, M$
    end for
    for $i = 1, \ldots, p$; $j \neq i$ **do**
        Sample $B_{ij} \mid E_{ij}, \mathbf{Y}, \mathbf{Z}_i$ from spike-and-slab
    end for
    Compute $\boldsymbol{\epsilon} = (I - B)^{-1}\mathbf{Y}$
    Sample $\gamma_1 \mid B, E$ from Inverse-Gamma
    for $i = 1, \ldots, p$ **do**
        Sample $Z_i^{(q)} \mid \boldsymbol{\epsilon}_i, \boldsymbol{\pi}_i, \boldsymbol{\mu}_i, \boldsymbol{\tau}_i$ from Categorical, $q = 1, \ldots, N$
        Sample $\boldsymbol{\pi}_i \mid \mathbf{Z}_i$ from Dirichlet
    end for
end for

Algorithm 2 BayesDCG: Bayesian Sparse Causal Discovery for DCGs

Initialize $\gamma, \gamma_1, E, B, \{\pi_i, \boldsymbol{\mu}_i, \boldsymbol{\tau}_i, \mathbf{Z}_i\}_{i=1}^p$
for $m = 1, \ldots, M_{\text{iter}}$ **do**
    Sample γ, γ_1 from Beta full conditionals
    Sample mixture parameters $\{\mathbf{Z}_i, \boldsymbol{\pi}_i, \boldsymbol{\mu}_i, \boldsymbol{\tau}_i\}_{i=1}^p$ as in Algorithm 1
    for each pair (i, j) with $i \neq j$ **do**
        // Joint edge and coefficient update
        Propose $E_{ij} = 1$ with $B_{ij} \sim N(0, \sigma^2_{\text{add}})$
        if $\rho(B) \geq 1$ **then** reject (stability check)
        else accept with MH ratio from joint posterior $[E, B \mid \mathbf{Y}]$
    end for
    for each (i, j) with $E_{ij} = 1$ **do**
        // Random walk refinement
        Propose $B'_{ij} \sim N(B_{ij}, \sigma^2_{\text{rw}})$
        if $\rho(B') < 1$ **then** accept with MH ratio
    end for
end for

5 Design and Implementation

The package `cyclinbayes` provides two MCMC samplers and three summary functions, all implemented in Rcpp for efficiency.

Samplers

- `BayesDAG()`: Bayesian LiNGAM sampler using collapsed Gibbs updates for the graph structure with simulated annealing. Returns posterior draws of the adjacency matrix

E, causal effect matrix B, and mixture parameters.

- `BayesDCG()`: Bayesian sampler for directed cyclic graphs. Jointly updates (E, B) via Metropolis–Hastings with random-walk proposals. Output format matches `BayesDAG()`.

Summary Functions

- `posterior_interval_est()`: Computes HPD and equal-tailed credible intervals for model parameters.
- `point_est_graph()`: Returns a point estimate of the graph by selecting the posterior weighted medoid under a specified distance (SHD, SID, or custom).
- `posterior_network_motif()`: Computes the posterior probability that a target subgraph $\mathcal{G}$ is contained in the sampled graphs:

$$\hat{\pi} = \frac{1}{m} \sum_{t=1}^{m} I(\mathcal{G} \subseteq \mathcal{G}^{(\sqcup)}).$$

6 Examples

6.1 Acyclic Example (DAG)

This section demonstrates how to use `BayesDAG()` to recover causal effects and quantify posterior uncertainty. We will start with simulated data, where we know the true DAG structure, then fit the estimated model and compare it with the truth.

6.1.1 Running the Bayesian LiNGAM Sampler

The function `BayesDAG()` performs Bayesian inference for sparse DAG structures under a Gaussian mixture noise model. The user specifies a data matrix $X_{N\times p}$, hyperparameters, and the number of MCMC iterations:

```
set.seed(21)



########################################
# Simulation and MCMC settings
########################################
N = 300    # sample size
num_covariates = 10     # number of features
M = 2      # number of mixture components
num_iter = 10000  # number of MCMC iterations


########################################
# Hyperparameter setup
########################################
params = list(
  a_mu      = 0,
  b_mu      = 2,
  a_gamma   = 0.5,
  b_gamma   = 0.5,
  a_gamma_1 = 2,
  b_gamma_1 = 1,
  a_tao     = 2,
  b_tao     = 1,
  a_og_tao  = 0.01,
  b_og_tao  = 0.01,
  alpha     = 1
)
```

```
########################################
# Run Bayesian LiNGAM (DAG) sampler
########################################
results_lists = BayesDAG(
  data_matrix,
  params$a_mu,
  params$b_mu,
  params$a_gamma,
  params$b_gamma,
  params$a_tao,
  params$b_tao,
  params$a_og_tao,
  params$b_og_tao,
  params$a_gamma_1,
  params$b_gamma_1,
  params$alpha,
  M,
  num_iter
)
```

The outputs consist of posterior draws of adjacency matrices, causal effect matrices, mixture parameters (π, μ, τ), and sparsity parameters γ and γ_1.

6.1.2 Selecting a posterior graph

We summarize the sampled adjacency matrices using the *posterior weighted medoid*, i.e., the graph that minimizes the posterior-weighted average distance to all sampled graphs. For DAGs, both structural Hamming distance (`SHD`) and structural intervention distance (`SID`)

are available.

Dependency note (SID). The `SID` distance is computed using the `SID` package. On some systems, `SID` also depends on the Bioconductor packages `graph` and `RBGL`. If you encounter a missing `RBGL` or `graph` error, install them via `BiocManager::install(c("graph","RBGL"))`.

```
Adjacency_est_shd <- point_est_graph(Adjacency_matrix_list,
  dist_type = "shd")
Adjacency_est_sid <- point_est_graph(Adjacency_matrix_list,
  dist_type = "sid")
```

Alternatively, using a user-defined distance function, we can compute the posterior weighted medoid with respect to that custom metric.

```
custom_edge_mismatch <- function(A, B) sum(abs(A - B))

Adjacency_est_custom <- point_est_graph(Adjacency_matrix_list,
  dist_type = "custom",
  dist_fun = custom_edge_mismatch)
```

All three often coincide when the posterior is concentrated.

6.1.3 Visualizing the True DAG

We visualize the ground truth graph and the estimated graphs for comparison with posterior results:

```
library(igraph)

g_true = graph_from_adjacency_matrix(Adjacency_matrix_true,
mode = "directed")
g_est  = graph_from_adjacency_matrix(Adjacency_est_shd,
```

```
mode = "directed")


par(mfrow=c(1,2))
plot(g_true, main = "True DAG")
plot(g_est,  main = "Estimated DAG (SHD)")
```

This provides an immediate qualitative comparison between the ground truth and the inferred causal structure.

6.1.4 Uncertainty Quantification for the Graph Structure

The function `posterior_network_motif()` quantifies uncertainty by evaluating the posterior mass (i.e., relative frequency) of a candidate (sub)graph by checking, for each sampled adjacency matrix, whether all edges in the candidate graph are present in that posterior draw.

As an example, let `Adjacency_matrix_true` indicate the true graph and let `Adjacency_matrix_list` be the posterior adjacency samples returned by `Bayes_DAG()`. The posterior mass assigned to the true graph can be computed as follows:

```
true_graph_structure = igraph::graph_from_adjacency_matrix(
Adjacency_matrix_true)


posterior_network_motif(true_graph_structure,
Adjacency_matrix_list)
```

This provides a quantitative measure of how strongly the posterior distribution supports a particular graph.

6.1.5 Posterior Interval Estimates

The function `posterior_interval_est()` computes both equal-tailed credible intervals and higest posterior density (HPD) intervals for model parameters at a user-specified level, returning interval bounds for each parameter entry.

```
Causal_summary = posterior_interval_est(Causal_effect_matrix_list,
level = 0.95)
hpd_matrix = Causal_summary$hpd_matrix
ci_matrix = Causal_summary$ci_matrix
```

Then to plot the HPD and credible intervals for non zero causal effects

```
########################################
# Extracting nonzero HPD intervals
########################################
par(mfrow=c(2,1))

nonzero_cols = which(colSums(hpd_matrix_acyclic) != 0)
num_non_zero_coef = length(nonzero_cols)
data_1 = data.frame(cbind(1:num_non_zero_coef,
t(hpd_matrix_acyclic[,which(colSums(hpd_matrix_acyclic)!=0)])))

nonzero_cols = which(colSums(hpd_matrix_acyclic) != 0)

# subset and transpose so each row = coefficient
hpd_sub = t(hpd_matrix_acyclic[, nonzero_cols, drop = FALSE])
colnames(hpd_sub) = c("lower", "upper")
# row1 = lower, row2 = upper

data_1 = as.data.frame(hpd_sub)
```

```
data_1$x = factor(seq_len(nrow(data_1)))


data_1$mid = (data_1$lower + data_1$upper) / 2



ggplot(data_1, aes(x = x, y = mid)) +
  geom_point(size = 3) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.2) +
  labs(y = "Causal Weight Estimate with HPD Interval",
  x = "Nonzero causal
  effect coefficient (index)") + theme_minimal()



########################################
# Credible intervals
########################################
data_2 = data.frame(ci_matrix_acyclic[which(rowSums
(ci_matrix_acyclic)!=0),])
x = factor(1:nrow(data_2))
data_2 = cbind(x,data_2)


ggplot(data_2, aes(x = x, y = X2)) +
  geom_point(size = 3) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
  geom_errorbar(aes(ymin = X1, ymax = X3), width = 0.2) +
  labs(y = "Causal Weight Estimate with 95% CI", x = "Nonzero
causal effect
```

```
  coefficient (index)") +
  theme_minimal()
```

6.2 Cyclic Example (DCG)

The workflow for DCGs parallels the acyclic case: after running the sampler, we summarize posterior structure, inspect uncertainty in key parameters, and visualize the estimated graph.

6.2.1 Running DCG Sampler

We illustrate how to run the Bayesian DCG sampler on data stored in a matrix `data_matrix` with N samples and p variables. The hyperparameters controlling sparsity, mixture noise, and edge effect scaling are identical in form to those used for the DAG sampler.

```
########################################
Simulation and MCMC settings
########################################
N = 250 # Sample size for the test data
num_covariates = 7 # Number of features for test data
M = 2 # Number of finite clusters for mixed normal in likelihood
num_iter = 10000 # Number of iterations MCMC runs


########################################
Hyperparameter setup
########################################
params = list(
  a_mu = 0,
  b_mu = 2,
  a_gamma = 2,
  b_gamma = 1,
```

```
  a_gamma_1 = 2,
  b_gamma_1 = 1,
  a_tao = 2,
  b_tao = 1,
  alpha = 1
)

########################################
Fit Bayesian DCG model
########################################
results_list = BayesDCG(
data_matrix,
params$a_mu, params$b_mu,
params$a_gamma, params$b_gamma,
params$a_tao, params$b_tao,
params$a_gamma_1, params$b_gamma_1,
params$alpha,
M,
num_iter
)
```

The cyclic sampler similar to the acyclic case returns posterior draws of all model parameters, including adjacency matrices, causal effect matrices, mixture weights, and latent cluster allocations.

6.2.2 Posterior Analysis and Uncertainty Quantification for DCGs

Posterior summaries in the cyclic setting follow the same workflow as in the acyclic (DAG) case. After running the DCG sampler `BayesDCG()`, users can quantify uncertainty in any pa-

rameter using `posterior_interval_est()` and graph structure using `posterior_network_motif()`. Posterior graph selection is performed with `point_est_graph()`, which identifies the weighted posterior medoid under a chosen distance metric. The only technical distinction is that SID is only valid for acyclic graphs; therefore, SHD (or a user-defined distance) should be used when selecting representative structures for general directed cyclic graphs.

7 Simulation Study

We look at the user interface and the performance of `BayesDAG` and `BayesDCG` through two different examples. In the first example for `BayesDAG`, we generate simulation error terms $\epsilon_i^{(q)}$ from three distributions, $i = 1 \ldots, p$ and $q = 1 \ldots, N$: (1) a finite mixture model $\sum_{k=1}^{M} \pi_{ik} N(\mu_{ik}, \tau_{ik})$, where $M = 2$, $(\mu_{i1}, \mu_{i2}) = (-0.5, 0.5)$, and $(\tau_{i1}, \tau_{i2}) = (0.1, 0.3)$, and $(\pi_{i1}, \pi_{i2}) = (0.5, 0.5)$, (2) t-distribution with degree of freedom (df) 7, and (3) Laplace distribution with location parameter $\mu = 0$ and scale parameter $b = 0.25$. To create the true causal effect matrix, for each entry in the matrix, we sample either 0 or 1 based on sparsity probability $\Delta = 0.9$ until the matrix B represents the adjacency matrix of a DAG. In the `BayesDCG` example, we generate $\epsilon_i^{(q)}$ from the same three distributions. Then, to create the true causal effect matrix B, we sample either 0 or 1 based on the sparsity probability $\Delta = 0.9$. Then in order to guarantee the inverse of $I - B$, we constrain the spectral radius of causal effect matrix B. We calculate $\rho(B)$, the largest modulus of the eigenvalue of B. If $|\rho(B)| \geq 0.95$, we rescale the causal effect matrix to be $(0.95/\rho(B))B$, to guarantee the spectral radius is less than 0.95. For both examples, we use the generated causal effect matrix and perform the operation, $\epsilon_i^{(q)}$ from

$$\vec{Y_i} = (I - B)^{-1} \vec{\epsilon_i},$$

where $\vec{\epsilon_i} = (\epsilon_i^{(1)}, \ldots, \epsilon_i^{(N)})^T$ and $\vec{Y_i} = (Y_i^{(1)}, \ldots, Y_i^{(N)})^T$. In this section, we compare our first method in `BayesDAG`, with two existing methods, ICA-LiNGAM [Shimizu et al., 2006]

and Direct LiNGAM [Shimizu et al., 2011] for various simulation settings. In our method, we ran 100,000 MCMC samples with a 75 percent burn-in time. For our second method, we compare `BayesDCG` with the existing approach `MR.RGM` [Sarkar and Ni, 2025] using five evaluation metrics. We focus on `MR.RGM` as a benchmark because it provides a readily usable implementation for cyclic/reciprocal structure with uncertainty quantification, whereas CCD implementations primarily target Markov equivalence classes and available code is often unmaintained or not reliably runnable in current R/Python environments.

7.1 Evaluation Metrics

For each of our simulation settings varied by (N, p) and the distribution of error ϵ, we access the precision of our corresponding adjacency matrix the true positive rate (TPR), the false positive rate (FPR), precision, accuracy, recall, and F1 score of E, which are defined as follows:

$$
\begin{aligned}
TPR &= \frac{\sum I(E_{ij}=1, \hat{E}_{ij}=1)}{\sum I(E_{ij}=1, \hat{E}_{ij}=1) + \sum I(E_{ij}=1, \hat{E_{ij}}=0)}; \\
FPR &= \frac{\sum I(E_{ij}=0, \hat{E}_{ij}=1)}{\sum I(E_{ij}=0, \hat{E}_{ij}=1) + \sum I(E_{ij}=0, \hat{E}_{ij}=0)}; \\
Precision &= \frac{\sum I(E_{ij}=1, \hat{E}_{ij}=1)}{\sum I(E_{ij}=1, \hat{E}_{ij}=1) + \sum I(E_{ij}=0, \hat{E}_{ij}=1)}; \\
Recall &= \frac{\sum I(E_{ij}=1, \hat{E}_{ij}=1)}{\sum I(E_{ij}=1, \hat{E}_{ij}=1) + \sum I(E_{ij}=1, \hat{E}_{ij}=0)}; \\
Accuracy &= \frac{\sum I(E_{ij}=1, \hat{E}_{ij}=1) + \sum I(E_{ij}=0, \hat{E_{ij}}=0)}{\sum I(E_{ij}=1, \hat{E}_{ij}=1) + \sum I(E_{ij}=1, \hat{E}_{ij}=0) + \sum I(E_{ij}=0, \hat{E_{ij}}=0) + \sum I(E_{ij}=0, \hat{E}_{ij}=1)}; \\
F1 &= 2\frac{Precision \cdot Recall}{Precision + Recall},
\end{aligned}
$$

where $\hat{E}_{ij}$ and E_{ij} represent the estimated adjacency matrix and the true adjacency matrix, respectively.

7.2 Simulation Results

As shown in Tables 1 and 2, the advantage of `BayesDAG` becomes increasingly pronounced as the feature dimension grows. In particular, when p increases to 30 and especially $p = 40$, `BayesDAG` consistently attains substantially higher F1 scores than both ICA-LiNGAM and Direct-LiNGAM, indicating markedly better performance in higher-dimensional settings. Other methods can achieve slightly higher recall by selecting more edges, but as p increases, this strategy typically produces many more false positives and substantially lowers precision. In contrast, `BayesDAG` maintains higher precision and a more favorable balance between false discoveries and missed edges, yielding more reliable graph recovery in high dimensions.

As shown in Tables 4, 5, and 6, `BayesDCG` consistently outperforms the RGM baseline, with the improvement becoming more pronounced as the dimension increases from $p = 20$ to $p = 40$. This widening gap reflects `BayesDCG`'s stronger balance between precision and recall in high-dimensional cyclic settings. By combining sparsity-inducing priors with stability-constrained MCMC updates, `BayesDCG` remains computationally tractable and better controls false positives even when the number of possible edges grows rapidly with p. Performance also improves with larger sample sizes, as posterior uncertainty concentrates around the true cyclic structure and associated causal effects, leading to more accurate and stable graph recovery.

8 Conclusion

We introduce `cyclinbayes`, an R package that provides a unified Bayesian framework for causal discovery under linear non-Gaussian structural equation models. The package implements two complementary methods: `BayesDAG` for acyclic graphs and `BayesDCG` for directed cyclic graphs with feedback. Both retain sparsity through spike-and-slab priors, capture heterogeneous noise via finite mixture error models, and use hybrid MCMC schemes for efficient

posterior inference.

Beyond estimating a single graph, `cyclinbayes` delivers full posterior uncertainty quantification for direct causal effects and user-specified network motifs—functionality largely absent from existing implementations—and offers one of the few software options for linear non-Gaussian DCG learning, with Bayesian uncertainty quantification in the cyclic setting. Finally, we propose a decision-theoretic posterior graph summarization approach that selects a representative structure by minimizing posterior expected loss under global discrepancy metrics such as SHD and SID. Together with an optimized Rcpp implementation, these features make `cyclinbayes` a practical toolbox for applied researchers seeking flexible causal modeling with rigorous uncertainty quantification.

References

Markus Kalisch, Martin Mächler, Diego Colombo, Marloes H. Maathuis, and Peter Bühlmann. Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, 47(11):1–26, 2012. 10.18637/jss.v047.i11.

Genta Kikuchi. rlingam: R implementation of lingam algorithms. *GitHub repository*, 2025. URL `https://github.com/gkikuchi/rlingam`.

Gustavo Lacerda, Peter Spirtes, Joseph Ramsey, and Patrik O. Hoyer. Discovering cyclic causal models by independent components analysis. *arXiv preprint arXiv:1206.3273*, 2012. URL `https://arxiv.org/abs/1206.3273`.

G. J. McLachlan and D. Peel. Finite mixture models. *Wiley Series in Probability and Statistics*, 2004.

Bitan Sarkar and Yang Ni. MR.RGM: an R package for fitting bayesian multivariate bidirec-

tional mendelian randomization networks. *Bioinformatics*, 41(4):btaf130, 2025. 10.1093/bioinformatics/btaf130. URL https://doi.org/10.1093/bioinformatics/btaf130.

Shohei Shimizu, Patrik O. Hoyer, Aapo Hyvärinen, and Antti Kerminen. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7: 2003–2030, 2006.

Shohei Shimizu, Takanori Inazumi, Yasuhiro Sogawa, Aapo Hyvärinen, Yoshinobu Kawahara, Takashi Washio, Patrik O. Hoyer, and Kenneth Bollen. DirectLiNGAM: A direct method for learning a linear non-gaussian structural equation model. *Journal of Machine Learning Research*, 12:1225–1248, 2011.

APPENDIX

This appendix consists of two sections: Sections A and B contain derivations of full conditional distributions under DAG and under DCG, respectively.

A Derivation of Full Conditional Distributions Under DAG

In this section, we derive the full conditional distribution of each parameter in the collapsed Gibbs sampler for DAG.

A.1 Full Conditional distribution of γ

$$\begin{aligned}
[\gamma|-] &\propto [\gamma][E_{21}|\gamma]...[E_{pp-1}|\gamma] \\
&= \gamma^{a_\gamma+\sum_{j=2}^{p}\sum_{l\in pa(j)}E_{jl}-1}(1-\gamma)^{b_\gamma+\sum_{j=2}^{p}\sum_{l\in pa(j)}(1-E_{jl})-1} \\
&\sim Beta\left(a_\gamma+\sum_{j=2}^{p}\sum_{l\in pa(j)}E_{jl}, b_\gamma+\sum_{j=2}^{p}\sum_{l\in pa(j)}(1-E_{jl})\right)
\end{aligned} \tag{γ}$$

A.2 Full Conditional distribution of γ_1

$$
\begin{aligned}
[\gamma_1|-] &\propto [\gamma_1][B_{21}|E_{21},\gamma_1]...[B_{pp-1}|E_{pp-1},\gamma_1] \\
&= \frac{b_\gamma^{a_\gamma}}{\Gamma(a_\gamma)}(\gamma_1)^{-(a_\gamma+1)}\exp\left(\frac{-b_\gamma}{\gamma_1}\right) \times \prod_{i=2,...,p,\, j\in pa(i)} \{I(B_{ij}=0)\}^{1-E_{ij}}\{N(0,\gamma_1)\}^{E_{ij}} \\
&= \frac{b_\gamma^{a_\gamma}}{\Gamma(a_\gamma)}(\gamma_1)^{-(a_\gamma+1)}\exp\left(\frac{-b_\gamma}{\gamma_1}\right) \\
&\times \prod_{i=2,...,p,\, j\in pa(i)} \{I(B_{ij}=0)\}^{1-E_{ij}}\{(\frac{1}{\sqrt{2\pi\gamma_1}})^{E_{ij}}\exp\left(-\frac{E_{ij}}{2\gamma_1}(B_{ij})^2\right)\} \qquad (\gamma_1) \\
&\propto (\gamma_1)^{-(a_\gamma+1)}\exp\left(\frac{-b_\gamma}{\gamma_1}\right)\prod_{i=2,...,p,\, j\in pa(i)}\exp\left(-\frac{E_{ij}}{2\gamma_1}B_{ij}^2\right)(\frac{1}{\sqrt{2\pi\gamma_1}})^{E_{il}} \\
&\times (\gamma_1)^{-(a_\gamma+\frac{\sum_{i=2}^p\sum_{j\in pa(i)}E_{ij}}{2}+1)}\exp\left(-\frac{b_\gamma+\frac{1}{2}\sum_{i=2}^p\sum_{j\in pa(i)}(E_{ij})(B_{ij})^2}{\gamma_1}\right) \\
&\sim IG\left(a_\gamma+\frac{\sum_{i=2}^p\sum_{j\in pa(i)}E_{ij}}{2}, b_\gamma+\frac{1}{2}\sum_{i=2}^p\sum_{j\in pa(i)}(E_{ij})(B_{ij})^2\right)
\end{aligned}
$$

A.3 Full conditional distribution of $Z_i^{(q)}$

$$
\begin{aligned}
&[Z_i^{(q)}|\,\pi_{i1},...,\pi_{iM}][\epsilon_1^{(1)},\cdots,\epsilon_p^{(1)},\cdots,\epsilon_1^{(q)},\cdots,\epsilon_p^{(N)}|all\,\pi, all\,\mu, all\,\tau] \\
&\propto \pi_{i1}^{I(Z_i^{(q)}=1)}...\pi_{iG}^{I(Z_i^{(q)}=M)}\prod_{k=1}^M\{N(\epsilon_i^{(q)},\mu_{ik},\tau_{ik})\}^{I(Z_i^{(q)}=k)} \qquad (Z_i^{(q)}) \\
&\{\pi_{i1}\,N(\epsilon_i^{(q)},\mu_{i1},\tau_{i1})\}^{I(Z_i^{(q)}=1)}\cdots\{\pi_{jM}N(\epsilon_i^{(q)},\mu_{iM},\tau_{iM})\}^{I(Z_i^{(q)}=M)} \\
&\sim Categorical\left(\pi_{i1}N(\epsilon_i^{(q)},\mu_{i1},\tau_{i1}),\cdots,\pi_{iM}N(\epsilon_i^{(q)},\mu_{iM},\tau_{iM})\right)
\end{aligned}
$$

A.4 Full conditional distribution of π_{ik}

$$
\begin{aligned}
&[\vec{\pi_i}][\mathbf{Z}_i|\vec{\pi_i}] \propto \prod_{q=1}^N\prod_{k=1}^M(\pi_{ik})^{\alpha-1}\{(\pi_{i1})^{I(Z_i^{(q)}=1)}...(\pi_{iM})^{I(Z_j^{(q)}=M)}\} \qquad (\pi_{ik}) \\
&\pi_{i1},...,\pi_{ik},...,\pi_{iM} \sim Dir(\alpha+\sum_{q=1}^N I(Z_i^{(q)}=1),...,\alpha+\sum_{q=1}^N I(Z_i^{(q)}=M))
\end{aligned}
$$

A.5 Full conditional distribution of μ_{ik}

$$
\begin{aligned}
&[\mu_{ik}][\epsilon_1^{(1)}, \cdots, \epsilon_p^{(1)}, \cdots, \epsilon_1^{(N)}, \cdots, \epsilon_p^{(N)} \mid all\,\pi, all\,\mu, all\,\tau] \\
&\quad \propto \frac{1}{\sqrt{2\pi b_\mu}} \exp\left(-\frac{1}{2b_\mu}(\mu_{ik} - a_\mu)^2\right) \prod_{q=1}^{N} \prod_{i'=1}^{p} \prod_{k'=1}^{G} N(\epsilon_i^{(q)}, \mu_{i'k'}, \tau_{jk})^{I(Z_i^{(q)}=k')} \\
&\quad = \frac{1}{\sqrt{2\pi b_\mu}} \exp\left(-\frac{1}{2b_\mu}(\mu_{ik} - a_\mu)^2\right) \prod_{q=1}^{N} \{N(\epsilon_i^{(q)}, \mu_{ik}, \tau_{ik})\}^{I(Z_i^{(q)}=k)} \\
&\quad \propto \exp\left(-\frac{1}{2b_\mu}((\mu_{ik})^2 - 2a_\mu\mu_{ik})\right) \prod_{q=1}^{N} \left\{\left(\frac{1}{\sqrt{2\pi\tau_{ik}}}\right) \exp\left(-\frac{1}{2\tau_{ik}}(\epsilon_i^{(q)} - \mu_{ik})^2\right)\right\}^{I(Z_i^{(q)}=k)} \\
&\quad \propto \exp\left(-\frac{1}{2b_\mu}\{(\mu_{ik})^2 - 2a_\mu\mu_{ik}\}\right) \left(\frac{1}{\sqrt{2\pi\tau_{ik}}}\right)^{\sum_{q=1}^{N} I(Z_i^{(q)}=k)} \\
&\qquad \cdot \exp\left(-\sum_{q=1}^{N} \frac{I(Z_i^{(q)}=k)}{2\tau_{ik}}\{(\mu_{ik})^2 - 2\mu_{ik}\epsilon_i^{(q)}\}\right) \\
&\quad \propto \exp\left(\left\{-\frac{1}{2b_\mu} - \frac{\sum_{q=1}^{N} I(Z_i^{(q)}=k)}{2\tau_{ik}}\right\}(\mu_{ik})^2 - 2\left\{-\frac{1}{2b_\mu}a_\mu\mu_{ik} - \sum_{q=1}^{N}\frac{I(Z_i^{(q)}=k)}{2\tau_{ik}}\mu_{ik}\epsilon_i^{(q)}\right\}\right) \\
&\quad = \exp\left(-\frac{1}{2}\left\{\frac{1}{b_\mu} + \frac{\sum_{q=1}^{N} I(Z_i^{(q)}=k)}{\tau_{ik}}\right\}(\mu_{ik})^2 + \left\{\frac{a_\mu}{b_\mu} + \sum_{q=1}^{N}\frac{I(Z_i^{(q)}=k)}{\tau_{ik}}\epsilon_i^{(q)}\right\}\mu_{ik}\right) \\
&\quad = \exp\left(-\frac{1}{2}\left\{\frac{1}{b_\mu} + \frac{\sum_{q=1}^{N} I(Z_i^{(q)}=k)}{\tau_{ik}}\right\}\left\{(\mu_{ik})^2 - 2\frac{\frac{a_\mu}{b_\mu} + \sum_{q=1}^{N}\frac{I(Z_i^{(q)}=k)}{\tau_{ik}}\epsilon_i^{(q)}}{\frac{1}{b_\mu} + \frac{\sum_{q=1}^{N} I(Z_i^{(q)}=k)}{\tau_{ik}}}\mu_{ik}\right\}\right) \\
&\quad \propto \exp\left(-\frac{1}{2}\left\{\frac{1}{b_\mu} + \frac{\sum_{q=1}^{N} I(Z_i^{(q)}=k)}{\tau_{ik}}\right\}\left\{\mu_{ik} - \frac{\frac{a_\mu}{b_\mu} + \sum_{q=1}^{N}\frac{I(Z_i^{(q)}=k)}{\tau_{ik}}\epsilon_i^{(q)}}{\frac{1}{b_\mu} + \frac{\sum_{q=1}^{N} I(Z_i^{(q)}=k)}{\tau_{ik}}}\right\}^2\right) \\
&\quad \sim N\left(\frac{\frac{a_\mu}{b_\mu} + \sum_{q=1}^{N}\frac{I(Z_i^{(q)}=k)}{\tau_{ik}}\epsilon_i^{(q)}}{\frac{1}{b_\mu} + \frac{\sum_{q=1}^{N} I(Z_i^{(q)}=k)}{\tau_{ik}}},\ \frac{1}{\frac{1}{b_\mu} + \frac{\sum_{q=1}^{N} I(Z_i^{(q)}=k)}{\tau_{ik}}}\right)
\end{aligned}
$$

(μ_{ik})

A.6 Full conditional distribution of τ_{ik}

$$
\begin{aligned}
&[\tau_{ik}][\epsilon_1^{(1)}, \cdots, \epsilon_p^{(1)}, \cdots, \epsilon_1^{(N)}, \cdots, \epsilon_p^{(N)} \mid all\ \pi, all\ \mu, all\ \tau] \\
&\quad \propto \tau_{ik}^{-(a_\tau+1)} \exp\left(-\frac{b_\tau}{\tau_{ik}}\right) \prod_{q=1}^{N} \prod_{i'=1}^{p} \prod_{k'=1}^{M} N(\epsilon_i^{(q)}, \mu_{i'k'}, \tau_{i'k'})^{I(Z_j^{(q)}=k')} \\
&\quad \propto \tau_{ik}^{-(a_\tau+1)} \exp\left(-\frac{b_\tau}{\tau_{ik}}\right) \prod_{q=1}^{N} N(\epsilon_i^{(q)}, \mu_{ik}, \tau_{ik})^{I(Z_i^{(q)}=k)} \\
&\quad \propto \tau_{ik}^{-(a_\tau+1)} \exp\left(-\frac{b_\tau}{\tau_{ik}}\right) \prod_{q=1}^{N} \left\{ \left(\frac{1}{\sqrt{2\pi\tau_{ik}}}\right) \exp\left(-\frac{1}{2\tau_{ik}}(\epsilon_i^{(q)} - \mu_{ik})^2\right) \right\}^{I(Z_i^{(q)}=k)} \\
&\quad \propto \tau_{ik}^{-\left(a_\tau + \frac{\sum_{q=1}^{N} I(Z_i^{(q)}=k)}{2} + 1\right)} \exp\left(-\frac{b_\tau}{\tau_{ik}} - \sum_{q=1}^{N} \frac{I(Z_i^{(q)}=k)}{2\tau_{ik}} (\epsilon_i^{(q)} - \mu_{ik})^2 \right) \\
&\quad \propto \tau_{ik}^{-\left(a_\tau + \frac{\sum_{q=1}^{N} I(Z_i^{(q)}=k)}{2} + 1\right)} \exp\left(-\frac{1}{\tau_{ik}} \left[b_\tau + \sum_{q=1}^{N} \frac{I(Z_i^{(q)}=k)}{2} (\epsilon_i^{(q)} - \mu_{ik})^2 \right] \right) \\
&[\tau_{ik} \mid -] \propto IG\left(a_\tau + \frac{\sum_{q=1}^{N} I(Z_i^{(q)}=k)}{2},\ b_\tau + \sum_{q=1}^{N} \frac{I(Z_i^{(q)}=k)}{2} (\epsilon_i^{(q)} - \mu_{ik})^2 \right)
\end{aligned}
\tag{τ_{ik}}
$$

A.7 Full conditional Distribution of B_{ij}

$$
\begin{aligned}
&[B_{ij} \mid E_{ij}, \gamma_1][y_i^{(1)}, \ldots, y_i^{(N)} \mid B_{i1}, \ldots, B_{ii-1}, B_{ii+1}, \ldots, B_{ip}, \epsilon_i, Z_i^{(q)}, \\
&\qquad\qquad y_1^{(1)}, \ldots, y_{i-1}^{(1)}, y_{i+1}^{(1)}, \ldots, y_p^{(1)}, \ldots, y_1^{(N)}, \ldots, y_{i-1}^{(N)}, y_{i+1}^{(N)}, \ldots, y_p^{(N)}] \\
&\propto [E_{ij}(N(0,\gamma_1)) + (1-E_{ij})(I(B_{ij}=0))] \left[\prod_{q=1}^{N} \prod_{k=1}^{M} \{N(\mu_{ik} + \sum_{j' \in pa(i)} B_{ij'} Y_{j'}^{(q)}, \tau_{ik})\}^{I(Z_i^{(q)}=k)} \right] \\
&\propto [E_{ij}(\exp(-\frac{E_{ij}}{2\gamma_1} B_{ij}^2)) + (1-E_{ij})(I(B_{ij}=0))] \\
&[\exp(-\sum_{q=1}^{N}\sum_{k=1}^{M} \frac{I(Z_i^{(q)}=k)}{2\tau_{ik}} \{y_i^{(q)} - (\mu_{ik} + \sum_{j' \in pa(i)} B_{ij'} Y_{j'}^{(q)})\}^2)] \\
&\propto [E_{ij}(\exp(-\frac{E_{ij}}{2\gamma_1} B_{ij}^2)) + (1-E_{ij})(I(B_{ij}=0))] \exp\Big(-\sum_{q=1}^{N}\sum_{k=1}^{M} \frac{I(Z_i^{(q)}=k)}{2\tau_{ik}} \{(y_i^{(q)})^2 \\
&\quad - 2y_i^{(q)}(\mu_{ik} + \sum_{j' \in pa(i)} B_{ij'} Y_{j'}^{(q)}) + (\mu_{ik} + \sum_{j' \in pa(i)} B_{ij'} Y_{j'}^{(q)})^2\}\Big) \\
&\propto [E_{ij}(\exp(-\frac{E_{ij}}{2\gamma_1} B_{ij}^2)) + (1-E_{ij})(I(B_{ij}=0))] \exp\Big(-\sum_{q=1}^{N}\sum_{k=1}^{M} \frac{I(Z_i^{(q)}=k)}{2\tau_{ik}} \{2\mu_{ik} \sum_{j' \in pa(i)} B_{ij'} Y_{j'}^{(q)} \\
&\quad + (\sum_{j' \in pa(i)} B_{ij'} Y_{j'}^{(q)})^2 - 2y_i^{(q)} \sum_{j' \in pa(i)} B_{ij'} Y_{j'}^{(q)})\}\Big) \\
&\propto [E_{ij}(\exp(-\frac{E_{ij}}{2\gamma_1} B_{ij}^2)) + (1-E_{ij})(I(B_{ij}=0))] \exp\Big(-\sum_{q=1}^{N}\sum_{k=1}^{M} \frac{I(Z_i^{(q)}=k)}{2\tau_{ik}} \{(B_{ij} Y_j^{(q)})^2 \\
&\quad + 2\mu_{ik} B_{ij} y_j^{(q)} + 2\sum_{j' \neq j} B_{ij} B_{ij'} Y_j^{(q)} Y_{j'}^{(q)} - 2B_{ij} y_i^{(q)} y_j^{(q)}\}\Big) \\
&= [E_{ij}(\exp(-\frac{E_{ij}}{2\gamma_1} B_{ij}^2)) + (1-E_{ij})(I(B_{ij}=0))] \exp\Big(\{-\sum_{q=1}^{N}\sum_{k=1}^{M} (y_j^{(q)})^2 \frac{I(Z_i^{(q)}=k)}{2\tau_{ik}}\} B_{ij}^2 \\
&\quad + 2B_{ij} \sum_{q=1}^{N}\sum_{k=1}^{M} y_j^{(q)} \frac{I(Z_i^{(q)}=k)}{2\tau_{ik}} (-\mu_{ik} - \sum_{j' \neq j} B_{ij'} Y_{j'}^{(q)} + y_i^{(q)})\Big).
\end{aligned}
\tag{B_{ij}}
$$

Note: since we need to obtain the pdf for B_{ij}, $\sum_{l \in pa(i), l \neq j} B_{il}$ can be considered as a constant. This means that $\exp(-\{(-2y_i^{(q)} \sum_{j' \in pa(i)} B_{ij'} Y_{j'}^{(q)})\}) = \exp(-\{(-2y_i^{(q)} B_{ij} Y_j^{(q)})\})$.

Therefore, if $E_{ij} = 0$, then $B_{ij} = 0$;

Else if $B_{ij} \neq 0$,

$$[B_{ij} \mid -] \propto N\left(\frac{\sum_{q=1}^{N}\sum_{k=1}^{M} y_j^{(q)} \frac{I(Z_i^{(q)}=k)}{\tau_{ik}}(y_i^{(q)} - (\mu_{ik} + \sum_{j' \neq j} B_{ij'} Y_{j'}^{(q)}))}{\frac{1}{\gamma_1} + \sum_{q=1}^{N}\sum_{k=1}^{M}(y_j^{(q)})^2 \frac{I(Z_i^{(q)}=k)}{\tau_{ik}}}, \frac{1}{\frac{1}{\gamma_1} + \sum_{q=1}^{N}\sum_{k=1}^{M}(y_j^{(q)})^2 \frac{I(Z_i^{(q)}=k)}{\tau_{ik}}}\right).$$

A.8 E Full Conditional

A.8.1 Marginal conditional Distribution of E by integrating out B

$$\begin{aligned}
[E_{21}, \ldots, E_{ij}, \ldots, E_{pp-1} \mid -] &\propto [Y_1^{(1)}, \cdots, Y_p^{(1)}, \cdots, Y_1^{(N)}, \cdots, Y_p^{(N)} \mid E_{21}, \cdots, E_{ij}, \cdots, E_{pp-1}] \times [E \mid \gamma] \\
&= \int\int \ldots \int \{[Y_1^{(1)}, \cdots, Y_p^{(1)}, \cdots, Y_1^{(N)}, \cdots, Y_p^{(N)} \mid B_{21}, ..., B_{pp-1}] \\
&\quad \times [B_{21} \mid E_{21}, \gamma_1] \ldots [B_{pp-1} \mid E_{pp-1}, \gamma_1]\}\, dB_{21} \ldots dB_{pp-1} \times [E_{21}, \ldots, E_{ij}, \ldots, E_{pp-1} \mid \gamma].
\end{aligned} \tag{E}$$

Let $Pa(i) = \cup_{l'=0}^{p-1}\{(j_1, \ldots, j_{l'}) : 1 \leq j_1 < \ldots < j_{l'} \leq p,\, j_m \neq i, \text{ for } m = 1, \cdots, l'\}$. For a given $(j_1, \ldots, j_l) \in Pa(i)$, we can re-express our model as follows:

$$\begin{aligned}
\vec{Y_i} = \begin{bmatrix} Y_i^{(1)} \\ Y_i^{(2)} \\ \vdots \\ Y_i^{(N)} \end{bmatrix} &= \begin{bmatrix} Y_{j_1}^{(1)} & Y_{j_2}^{(1)} & \ldots & Y_{j_l}^{(1)} \\ Y_{j_1}^{(2)} & Y_{j_2}^{(2)} & \ldots & Y_{j_l}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{j_1}^{(N)} & Y_{j_2}^{(N)} & \ldots & Y_{j_l}^{(N)} \end{bmatrix} \begin{bmatrix} B_{ij_1} \\ B_{ij_2} \\ \vdots \\ B_{ij_l} \end{bmatrix} + \begin{bmatrix} \epsilon_i^{(1)} \\ \epsilon_i^{(2)} \\ \vdots \\ \epsilon_i^{(N)} \end{bmatrix} \\
&= \begin{bmatrix} \vec{Y_{j_1}} & \vec{Y_{j_2}} & \ldots & \vec{Y_{j_l}} \end{bmatrix} \vec{B_i} + \vec{\epsilon_i} \\
&\sim D_i \vec{B_i} + MVN(\vec{\mu_i}, \vec{\tau_i} I) \quad \text{for } i = 1, \ldots, p \text{ and } D_i = (Y_{j_1}, \ldots, Y_{j_l}).
\end{aligned}$$

$$\begin{bmatrix} \epsilon_i^{(1)} \\ \epsilon_i^{(2)} \\ \vdots \\ \epsilon_i^{(N)} \end{bmatrix} \sim MVN\left(\begin{bmatrix} \mu_{ik_{(1)}} \\ \mu_{ik_{(2)}} \\ \vdots \\ \mu_{ik_{(N)}} \end{bmatrix}, \begin{bmatrix} \tau_{ik_{(1)}} \\ \tau_{ik_{(2)}} \\ \vdots \\ \tau_{ik_{(N)}} \end{bmatrix} I\right), \quad \text{where } k_{(q)} = 1, \ldots, G \text{ satisfies } Z_{ik}^{(q)} = 1 \text{ if } k = k_{(q)}.$$

The multivariate priors for the vector B are

$$\left[\begin{bmatrix} B_{ij_1} \\ B_{ij_2} \\ \vdots \\ B_{ij_l} \end{bmatrix} \middle\| \begin{bmatrix} E_{ij_1} \\ E_{ij_2} \\ \vdots \\ E_{ij_l} \end{bmatrix}, \gamma_1 \right] = [\vec{B}_i \mid \vec{E}_i, \gamma_1] \sim MVN(0, \gamma_1 I). \tag{1}$$

Then the marginal conditional distribution of the ith row of E by integrated the ith row of B is

$$\begin{aligned}
&[E_{i1}, \ldots, E_{ii-1}, E_{ii+1}, \ldots E_{ip} \mid -] \\
&\propto [Y_1^{(1)}, \cdots, Y_p^{(1)}, \cdots, Y_1^{(N)}, \cdots, Y_p^{(N)} \mid E_{i1}, \ldots, E_{ii-1}, E_{ii+1}, \ldots E_{ip}] \times [E_{i1}, \ldots, E_{ii-1}, E_{ii+1}, \ldots E_{ip} \mid \gamma] \\
&= \sum_{l=0}^{p-1} \sum_{(j_1,\ldots,j_l) \in Pa(i)} [\prod_{j' \neq j_1,\ldots,j_l} (1 - E_{ij'})][\prod_{m=1}^{l} E_{ij_m}] \\
&\quad \times \int \int \ldots \int \{[Y_{j_1}^{(1)}, \cdots, Y_{j_l}^{(1)}, \cdots, Y_{j_1}^{(q)}, \cdots, Y_{j_l}^{(q)}, \cdots, Y_{j_1}^{(N)}, \cdots, Y_{j_l}^{(N)} \mid B_{ij_1}, \ldots, B_{ij_l}] \\
&\quad \times [B_{ij_1} \mid E_{ij_1}, \gamma_1] \ldots [B_{ij_l} \mid E_{ij_l}, \gamma_1]\} \, dB_{ij_1} \ldots dB_{ij_l} \times [E_{i1}, \ldots, E_{ii-1}, E_{ii+1}, \ldots E_{ip} \mid \gamma] \\
&= \sum_{l=0}^{p-1} \sum_{(j_1,\ldots,j_l) \in Pa(i)} [\prod_{j' \neq j_1,\ldots,j_l} (1 - E_{ij'})][\prod_{m=1}^{l} E_{ij_m}] \\
&\quad \times \int \int \ldots \int \{[Y_1^{(1)}, \cdots, Y_p^{(1)}, \cdots, Y_1^{(N)}, \cdots, Y_p^{(N)} \mid \vec{B}_i] \\
&\quad \times [\vec{B}_i \mid \vec{E}_i, \gamma_1] d\vec{B}_i] \times [E_{i1}, \ldots, E_{ii-1}, E_{ii+1}, \ldots E_{ip} \mid \gamma] \\
&= \sum_{l=0}^{p-1} \sum_{(j_1,\ldots,j_l) \in Pa(i)} [\prod_{j' \neq j_1,\ldots,j_l} (1 - E_{ij'})][\prod_{m=1}^{l} E_{ij_m}] \\
&\quad \times \int \int \ldots \int \{[Y_1^{(1)}, \cdots, Y_p^{(1)}, \cdots, Y_1^{(N)}, \cdots, Y_p^{(N)} \mid \vec{B}_i] \\
&\quad \times [\vec{B}_i \mid \vec{E}_i, \gamma_1][\vec{E}_i \mid \gamma] d\vec{B}_i] \\
&= \sum_{l=0}^{p-1} \sum_{(j_1,\ldots,j_l) \in Pa(i)} [\prod_{j' \neq j_1,\ldots,j_l} (1 - E_{ij'})][\prod_{m=1}^{l} E_{ij_m}] \times \mathbf{\Gamma}^i_{j_1,\ldots,j_l},
\end{aligned}$$

where $(j_1, \ldots, j_l) \in Pa(i) = \cup_{l'=0}^{p-1} \{(j_1, \ldots, j_{l'}) : 1 \leq j_1 < \ldots < j_{l'} \leq p, \, j_m \neq i, \text{ for } m = 1, \cdots, l'\}$. Here, the detailed calculation of $\mathbf{I}^{\mathbf{i}}_{\mathbf{j_1},\ldots,\mathbf{j_l}}$ is described in Section A.9.

Let $\mathcal{I}$ denote the set of all possible configurations of $(E_{21}, \ldots, E_{p,p-1})$, such that

$$\mathcal{I} = \{(E_{21}, \ldots, E_{p,p-1}) \mid E_{ij} \in \{0,1\},\ 1 \leq i,j \leq p,\ i \neq j\}\,.$$

The total number of elements in $\mathcal{I}$ is $2^{p(p-1)}$. The probability that $(E_{21}, \ldots, E_{p,p-1} \mid -) = I_e$, for some $I_e \in \mathcal{I}$, is proportional to $\tilde{P}_{I_e}$, where $\tilde{P}_{I_e}$ is defined in equation (E).

Define

$$P_{I_e} = \frac{\tilde{P}_{I_e}}{\sum_{I_e \in \mathcal{I}} \tilde{P}_{I_e}}.$$

Then, $[(E_{21}, \ldots, E_{pp-1}) = I_e \mid -] \sim \text{Categorical}(p_{I_e} : I_e \in \mathcal{I})$ which has a closed form.

However, direct sampling via Gibbs sampling using this closed form is infeasible due to the size of $\mathcal{I}$, which grows exponentially as $p \to \infty$. Hence, instead of using a Gibbs sampler, we use MH for sampling E.

A.9 Calculation of $\mathbf{I}^{\mathbf{i}}_{\mathbf{j_1},\ldots,\mathbf{j_l}}$, r, and $\tilde{r}$

$$\begin{aligned}
\mathbf{I}^{\mathbf{i}}_{\mathbf{j_1},\ldots,\mathbf{j_l}} &= \int [\vec{Y}_i | \vec{B}_i, \vec{\mu}_i, \vec{\tau}_i][\vec{B}_i | \vec{E}_i, \gamma_1] d\vec{B}_i \\
&= \int (2\pi)^{-N/2} (\det |\vec{\tau}_i I|)^{-1/2} \times \exp\Big(-\frac{1}{2}(\vec{Y}_i - (\vec{\mu}_i + D_i\vec{B}_i))^T (\vec{\tau}_i I)^{-1} (\vec{Y}_i - (\vec{\mu}_i + D_i\vec{B}_i))\Big) \\
&\quad \times (2\pi)^{-l/2} (\det |\vec{\gamma_1} I|)^{-1/2} \exp\Big(-\frac{1}{2}\vec{B}_i^T (\gamma_1 I)^{-1} \vec{B}_i\Big) d\vec{B}_i \\
&= (2\pi)^{-(l+N)/2} (\det |\vec{\tau}_i I|)^{-1/2} (\det |\vec{\gamma_1} I|)^{-1/2} \\
&\quad \times \int \exp\Big(\underbrace{-\frac{1}{2}(\vec{Y}_i - (\vec{\mu}_i + D_i\vec{B}_i)^T (\vec{\tau}_i I)^{-1} (\vec{Y}_i - (\vec{\mu}_i + D_i\vec{B}_i) - \frac{1}{2}\vec{B}_i^T (\gamma_1 I)^{-1} \vec{B}_i}_{\circledast}\Big) d\vec{B}_i
\end{aligned}$$

We first set $V = (\gamma_1 I_{l\times l})^{-1} + D_i^T (\vec{\tau}_i I_{N\times N})^{-1} D_i$ for $\vec{\tau}_i I_{N\times N} = diag(\tau_{ik_{(1)}}, \ldots, \tau_{ik_{(N)}})$ and

calculate

$$
\begin{aligned}
\circledast &= -\frac{1}{2}(\vec{Y}_i - (\vec{\mu}_i + D_i\vec{B}_i))^T(\vec{\tau}_i I)^{-1}(\vec{Y} - (\vec{\mu}_i + D_i\vec{B}_i)) - \frac{1}{2}\vec{B}_i^T(\gamma_1 I)^{-1}\vec{B}_i \\
&= -\frac{1}{2}((\vec{Y}_i - \vec{\mu}_i)^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i)) - \frac{1}{2}(\vec{B}_i^T((\gamma_1 I)^{-1} + D_i^T(\vec{\tau}_i I)^{-1}D_i)\vec{B}_i \\
&- 2\vec{B}_i^T D_i^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i)) \\
&= -\frac{1}{2}((\vec{Y}_i - \vec{\mu}_i)^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i)) - \frac{1}{2}(\vec{B}_i^T V\vec{B}_i - 2\vec{B}_i^T VV^{-1}D_i^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i)) \\
&= -\frac{1}{2}((\vec{Y}_i - \vec{\mu}_i)^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i)) - \frac{1}{2}(\vec{B}_i^T V\vec{B}_i - 2\vec{B}_i^T VV^{-1}D_i^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i) \\
&+ \frac{1}{2}(V^{-1}D_i^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i))^T V(V^{-1}D_i^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i)) \\
&- \frac{1}{2}(V^{-1}D_i^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i))^T V(V^{-1}D_i^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i))) \\
&= -\frac{1}{2}((\vec{Y}_i - \vec{\mu}_i)^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i)) \\
&\quad - \frac{1}{2}((\vec{B}_i - V^{-1}D_i^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i))^T V(\vec{B}_i - V^{-1}D_i^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i))) \\
&\quad + \frac{1}{2}(V^{-1}D_i^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i))^T V(V^{-1}D_i^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i)).
\end{aligned}
$$

Now, when we do the integral of the $\vec{B}_i$, we have

$$
\begin{aligned}
\mathbf{I}^i_{j_1,\ldots,j_l} &= (2\pi)^{-(l+N)/2}(\det|\vec{\tau}_i I|)^{-1/2}(\det|\vec{\gamma_1} I|)^{-1/2}\exp\Big(-\frac{1}{2}((\vec{Y}_i - \vec{\mu}_i)^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i) \\
&+ \frac{1}{2}(V^{-1}D_i^T(\tau_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i))^T V(V^{-1}D_i^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i))\Big) \\
&\int \exp\Big(-\frac{1}{2}((\vec{B}_i - V^{-1}D_i^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i))^T V(\vec{B}_i - V^{-1}D_i^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i))\Big)d\vec{B}_i \\
&= (2\pi)^{-(l+N)/2}(\det|\vec{\tau}_i I|)^{-1/2}(\det|\vec{\gamma_1} I|)^{-1/2}\exp\Big(-\frac{1}{2}((\vec{Y}_i - \vec{\mu}_i)^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i) \\
&+ \frac{1}{2}(V^{-1}D_i^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i))^T V(V^{-1}D_i^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i)))\Big)(2\pi)^{l/2}(\det|V|)^{1/2} \\
&= (2\pi)^{-N/2}\gamma_1^{-l/2}\left(\prod_{q=1}^{N}\tau_{ik_{(q)}}^{-1/2}\right)(\det|V^{-1}|)^{1/2}\exp\Big(-\frac{1}{2}(\vec{Y}_i - \vec{\mu}_i)^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i) \\
&+ \frac{1}{2}(V_i^{-1}(D_i)^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i))^T V(V^{-1}(D_i^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i)))\Big) \\
&= (2\pi)^{-N/2}\gamma_1^{-l/2}\left(\prod_{q=1}^{N}\tau_{ik_{(q)}}^{-1/2}\right)(\det|V|)^{1/2}\exp\Big(-\frac{1}{2}(\vec{Y}_i - \vec{\mu}_i)^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i) \\
&+ \frac{1}{2}((D_i)^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i))^T(V^{-1})(D_i^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i - \vec{\mu}_i)))\Big).
\end{aligned}
$$

Then, we can have the following form

$$\mathbf{I}^i_{j_1,\ldots,j_l} = (2\pi)^{-N/2}\gamma_1^{-l/2}\left(\prod_{q=1}^{N}\tau_{ik_{(q)}}^{-1/2}\right)(\det|V^{-1}|)^{1/2}\exp\Big(-\frac{1}{2}(\vec{Y}_i-\vec{\mu}_i)^T\delta_{q\tilde{q}}(\vec{\tau}_i I_{N\times N})^{-1}(\vec{Y}_i-\vec{\mu}_i)$$
$$+\frac{1}{2}(V^{-1}(D_i^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i-\vec{\mu}_i))^T V(V^{-1}(D_i^T(\vec{\tau}_i I)^{-1}(\vec{Y}_i-\vec{\mu}_i)))\Big).$$

Remark:

We note that we can calculate r and $\tilde{r}$ using $\mathbf{I}^i_{j_1,\ldots,j_i}$. For given $(j_1,\ldots,j_l)\in Pa(i)$ when $E_{ij}=1$ and $j=j_m$, the ratio r can be calculated as follows:

$$\begin{aligned} r &= \frac{\mathbf{I}^i_{j_1,\ldots,j_{l_2}}\text{when } E_{ji}=1 \text{ and } j=j_m}{\mathbf{I}^i_{j_1,\ldots,j_{l_2}}\text{when } E_{ji}=0 \text{ and } j=j_m} \\ &= \frac{\mathbf{I}^i_{j_1,\ldots,j_{m-1},j_m,j_{m+1},\ldots,j_{l_2}}}{\mathbf{I}^i_{j_1,\ldots,j_{m-1},j_{m+1},\ldots,j_l}}. \end{aligned} \tag{r}$$

For given $(j_1,\ldots,j_{l_1})\in Pa(i)$ when $(E_{ij}=1 \text{ and } j=j_{m_1})$ and $(i_1,\ldots,i_{l_2})\in Pa(j)$ when $(E_{ji}=1 \text{ and } i=i_{m_2})$, $\tilde{r}$ can be calculated as the following:

$$\tilde{r} = \frac{\mathbf{I}^i_{j_1,\ldots,j_{m_1-1},j_{m_1},j_{m_1+1},\ldots,j_{l_1}}\times\mathbf{I}^j_{i_1\ldots,i_{m_2-1},i_{m_2+1},\ldots,i_{l_2}}}{\mathbf{I}^i_{j_1,\ldots,j_{m_1-1},j_{m_1+1},\ldots,j_{l_1}}\times\mathbf{I}^j_{i_1\ldots,i_{m_2-1},i_{m_2},i_{m_2+1},\ldots,i_{l_2}}}. \tag{$\tilde{r}$}$$

A.10 MH enhanced with Two-Phase Simulated Annealing

To improve the MCMC sample for E_{ij} in MH, which was stuck in local maxima due to the binary value of E_{ij}, we enhance MH using two-phase simulated annealing.

We first define $\eta(m_c)$, a function of the m_cth MCMC iteration, which has the following properties.

Properties of $\eta(m_c)$

- $\eta(0)=1$;
- $\eta(\frac{N_I}{2})=0$, where N_I is the total number of iterations at MCMC;
- $\eta(m_c)$ is decreasing in m_c

Such a function can be chosen as $\eta(m_c) = e^{-c_1 \frac{m_c}{N_I}}$ that satisfies these properties. It is decreasing and $\eta(0) = 1$. We choose c_1 so that $\eta(\frac{N_I}{2}) \approx 0$.

So, next, we define the following quantity by adapting the idea of simulated annealing:

$$\begin{aligned}\mathbf{I}^{\mathbf{i(m_c)}}_{\mathbf{j_1},\ldots,\mathbf{j_l}} &= \mathbf{I}^{i}_{j_1,\ldots,j_l}\left((2\pi)^{-(l+G)}(\det|\vec{\tau}_i I|)^{-1/2}(\det|\Sigma|)^{-1/2}\mathbf{I}^{i}_{j_1,\ldots,j_l}\right)^{\eta(m_c)} \\ &= \Big(\underbrace{(2\pi)^{-(l+G)}}_{\text{Ⓐ}}\underbrace{(\det|\vec{\tau}_i I|)^{1/2}(\det|\Sigma|)^{1/2}}_{\text{Ⓑ}}\Big)^{\eta(m_c)}\left(\mathbf{I}^{i}_{j_1,\ldots,j_l}\right)^{1+\eta(m_c)},\end{aligned}$$

where

- Ⓐ $= (2\pi)^{-(l+G)}$ is a normalizing constant factor before the integration w.r.t. μ;
- Ⓑ $= (\det|\vec{\tau}_i I|)^{1/2}(\det|\Sigma|)^{1/2}$ is a normalizing constant factor in $\mathbf{I}^{i}_{j_1,\ldots,j_l}$.

We can easily observe that

$$\mathbf{I}^{\mathbf{i(m_c)}}_{\mathbf{j_1},\ldots,\mathbf{j_l}} \approx \mathbf{I}^{\mathbf{i}}_{\mathbf{j_1},\ldots,\mathbf{j_l}}$$

for $\frac{N_I}{2} \le m_c \le N_I$ since $\eta(\frac{N_I}{2}) \approx 10^{-8} \approx 0$.

For updating E_{ij} being 1 or 0, we consider M-H algorithm using this relation between $\mathbf{I}^{i(m_c)}$ and $\mathbf{I}^{i}$:

$$r = \frac{[Y_1,...,Y_p|E_{21},\ldots,E_{ij}=1,...,E_{pp-1}]}{[Y_1,...,Y_p|E_{21},\ldots,E_{ij}=0,...,E_{pp-1}]}$$

and

$$\begin{aligned} r_{m_c} &= \frac{\mathbf{I}^{\mathbf{i(m_c)}}_{\mathbf{j_1},\ldots,\mathbf{j_l}} \text{ when } E_{ij}=1}{\mathbf{I}^{\mathbf{i(m_c)}}_{\mathbf{j_1},\ldots,\mathbf{j_l}} \text{ when } E_{ij}=0} \\ &= \left(\frac{\gamma_1}{2\pi}\right)^{\eta(m_c)}\left(\frac{\mathbf{I}^{\mathbf{i}}_{\mathbf{j_1},\ldots,\mathbf{j_l}} \text{ when } E_{ij}=1}{\mathbf{I}^{\mathbf{i}}_{\mathbf{j_1},\ldots,\mathbf{j_l}} \text{ when } E_{ij}=0}\right)^{1+\eta(m_c)} \\ &= \left(\frac{\gamma_1}{2\pi}\right)^{\eta(m_c)}\left(\frac{[Y_1,...,Y_p|E_{21},\ldots,E_{ij}=1,...,E_{pp-1}]}{[Y_1,...,Y_p|E_{21},\ldots,E_{ij}=0,...,E_{pp-1}]}\right)^{1+\eta(m_c)} \\ &= \left(\frac{\gamma_1}{2\pi}\right)^{\eta(m_c)}(r)^{1+\eta(m_c)}. \end{aligned}$$

As a result, we can have the following facts:

$$r_{m_c} = \begin{cases} (\frac{\gamma_1}{2\pi})^{\eta(m_c)}(r)^{1+\eta(m_c)}, & \text{for } m_c < \frac{N_I}{2}, \\ r, & \text{for } m_c \geq \frac{N_I}{2}. \end{cases}$$

Interpretation of this function can be explained in term of two phases as described as follows;

- Phase 1 (adaptive phase): Acceptance ratio evolves according to $\eta(m_c)$, allowing either stricter or more exploratory behavior depending on $\gamma_1/(2\pi)$.
- Phase 2 (frozen phase): Acceptance ratio is fixed at r, producing a randomized but stable search.

We have an adaptive phase until half of the total MCMC samples. After that, we have the frozen phase. If the value of $r_{m_c} \geq 1$, then we consider that there is a node from $i \to j$. If there is no circle when $E_{ij} = 1$, then accept $E_{ij} = 1$; $E_{ji} = 1$ otherwise. If less than 1, there is a probability r_{m_c} that there is a node from $i \to j$. If $E_{ij} = 0$ with probability r_{m_c}, then accept $E_{ij} = 0$.

Next, if $E_{ij} = 1$ in the previous step, we then update $(E_{ij}, E_{ji}) = (1, 0)$ or (0,1). We define $\tilde{r}_{m_c}$ as

$$\tilde{r}_{m_c} = \begin{cases} (\tilde{r})^{1+\eta(m_c)}, & \text{for } m_c < \frac{N_I}{2}, \\ \tilde{r}, & \text{for } m_c \geq \frac{N_I}{2}. \end{cases}$$

If the value $\tilde{r}_{m_c} > 1$, then we accept that there is a node from $i \to j$ but not $j \to i$, that is $(E_{ij}, E_{ji}) = (1, 0)$. If less than 1, there is a probability $\tilde{r}_{m_c}$ that there is a node form $j \to i$ but not $i \to j$, that is $(E_{ij}, E_{ji}) = (0, 1)$.

B Derivation of Full Conditional Distribution Under DCG

The full conditional distributions for all parameters except (E, B) are identical to those provided in Appendix A. We therefore proceed to detail the sampling procedure from joint full conditional distribution $[E, B|-]$ in this section.

B.1 Adjacency and Causal Effect Matrix Update

The joint full conditional distribution of $[B, E|-]$ is

$$[B_{12}, \ldots, B_{pp-1}, E_{12}, \ldots, E_{pp-1}|-] \propto [\vec{Y}_1, \ldots, \vec{Y}_p|B_{12}, \ldots, B_{pp-1}, \vec{\mu}_1, \ldots, \vec{\mu}_p, \vec{\tau}_1, \ldots, \vec{\tau}_p]$$
$$[B_{12}|E_{12}, \gamma_1] \cdots [B_{pp-1}|E_{pp-1}, \gamma_1][E_{12}|\gamma] \cdots [E_{pp-1}|\gamma].$$

To update E_{ij} and B_{ij}, we assume that $E_{ij} = 1$ and from a proposal $v \sim N(0, 1)$, $B_{ij} = v$, then for a Metropolis Hastings ratio

$$r = \frac{[B_{12}, \ldots, B_{ij} = v, \ldots, B_{pp-1}, E_{12}, \ldots, E_{ij} = 1, \ldots, E_{pp-1}|-]}{[B_{12}, \ldots, B_{ij} = 0, \ldots, B_{pp-1}, E_{12}, \ldots, E_{ij} = 0, \ldots, E_{pp-1}|-]}.$$

Hence, the numerator of r can have the following expression:

$$[\vec{Y}_i|B_{12}, \ldots, B_{ij} = v, \ldots, B_{jp}, \vec{Y}_1, \ldots, \vec{Y}_{i-1}, \vec{Y}_{i+1}, \ldots \vec{Y}_p][B_{12}|E_{12}, \gamma_1][E_{12}|\gamma] \cdots [B_{ij} = v|E_{ij}, \gamma_1]$$
$$[E_{ij} = 1|\gamma] \cdots [B_{pp-1}|E_{pp-1}, \gamma_1][E_{pp-1}|\gamma]$$
$$= [D_i\vec{B}_i + MVN(\vec{\mu}_i, \vec{\tau}_i I)][N(B_{12}; 0, \gamma_1)] \cdots [N(B_{ij} = v; 0, \gamma_1)] \cdots [N(B_{pp-1}; 0, \gamma_1)]$$
$$\times \gamma^{k+1}(1 - \gamma)^{(p(p-1))-(k+1)}.$$

B.2 Metropolis Hastings Adjacency and Causal Update

For the cyclic sampler, each ordered pair (i, j) is updated via a joint Metropolis–Hastings birth/death move on (E_{ij}, B_{ij}). We use the spike–and–slab proposal kernel

$$q(E'_{ij}, B'_{ij} \mid E_{ij}, B_{ij}) = \mathbb{I}(E_{ij} = 0)\,\mathbb{I}(E'_{ij} = 1)\,\phi(B'_{ij}; 0, \sqrt{\gamma_1}) \;+\; \mathbb{I}(E_{ij} = 1)\,\mathbb{I}(E'_{ij} = 0)\,\delta_0(B'_{ij}),$$

where (E_{ij}, B_{ij}) denotes the current state and (E'_{ij}, B'_{ij}) the proposed state. In particular, if $E_{ij} = 0$, we propose adding the edge by drawing $B'_{ij} \sim N(0, \gamma_1)$. If $E_{ij} = 1$, we propose deleting the edge by setting $B'_{ij} = 0$.

B.2.1 Removing an edge $E_{ij} = 1 \to E'_{ij} = 0$

Let $P(B_{ij}, E_{ij} \mid -)$ denote the (unnormalized) full conditional of (B_{ij}, E_{ij}). Starting from a present edge $(E_{ij}, B_{ij}) = (1, B_{ij})$, we propose deletion by setting $(E'_{ij}, B'_{ij}) = (0, 0)$. The Metropolis–Hastings ratio is

$$r = \frac{P(B'_{ij}, E'_{ij} \mid -)}{P(B_{ij}, E_{ij} \mid -)} \cdot \frac{q(E_{ij}, B_{ij} \mid E'_{ij}, B'_{ij})}{q(E'_{ij}, B'_{ij} \mid E_{ij}, B_{ij})}.$$

Under the birth/death proposal, the forward proposal is the following,

$$q(E'_{ij}, B'_{ij} \mid E_{ij}, B_{ij}) = \mathbb{I}(E'_{ij} = 0)\,\delta_0(B'_{ij}),$$

while the reverse proposal corresponds to re-adding the edge with a slab draw,

$$q(E_{ij}, B_{ij} \mid E'_{ij}, B'_{ij}) = \mathbb{I}(E_{ij} = 1)\,\phi(B_{ij}; 0, \sqrt{\gamma_1}).$$

Hence, the proposal ratio contributes the factor $\phi(B_{ij}; 0, \sqrt{\gamma_1})$.

B.2.2 Adding an edge $E_{ij} = 0 \to E'_{ij} = 1$

Starting from an absent edge $(E_{ij}, B_{ij}) = (0, 0)$, we propose addition by drawing $B'_{ij} \sim N(0, \gamma_1)$ and setting $E'_{ij} = 1$. The Metropolis–Hastings ratio is

$$r = \frac{P(B'_{ij}, E'_{ij} \mid -)}{P(B_{ij}, E_{ij} \mid -)} \cdot \frac{q(E_{ij}, B_{ij} \mid E'_{ij}, B'_{ij})}{q(E'_{ij}, B'_{ij} \mid E_{ij}, B_{ij})}.$$

The forward proposal is the slab density,

$$q(E'_{ij}, B'_{ij} \mid E_{ij}, B_{ij}) = \mathbb{I}(E'_{ij} = 1)\, \phi(B'_{ij}; 0, \sqrt{\gamma_1}),$$

and the reverse proposal is deterministic deletion back to zero,

$$q(E_{ij}, B_{ij} \mid E'_{ij}, B'_{ij}) = \mathbb{I}(E_{ij} = 0)\, \delta_0(B_{ij}).$$

Thus, the proposal ratio contributes the factor $1/\phi(B'_{ij}; 0, \sqrt{\gamma_1})$.

B.2.3 Adaptive proposal variance for causal effect updates

For cyclic graphs, proposals that add an edge require sampling B'_{ij}. To stabilize mixing we create the following adaptive rule:

- During burn in (first 1000 iterations), we use a fixed proposal standard deviation at $\sigma_{add} = 0.15$.
- After burn in, the proposal standard deviation matches the slab scale at $\sigma_{add} = \sqrt{\gamma_1}$.

Therefore, the proposal is

$$B'_{ij} \sim N(0, \sigma^2_{add}). \tag{2}$$

The purpose of the adaptive proposal variance serves two purposes:

(i) The fixed 0.15 standard deviation encourages broad exploration and prevents early sticking.

(ii) After burn-in, matching σ_{add} to $\sqrt{\gamma_1}$ adapts the proposal to the posterior scale, improving mixing and acceptance rates.

Due to the more complex geometry of the posterior space of cyclic graphs, this ensures more robust exploration of both the adjacency and causal spaces.

B.3 Random Walk Update of Causal Weights

B.3.1 Adaptive random walk proposal variance schedule

To improve mixing across the MCMC run, we use an adaptive proposal variance that gradually increases from a small exploratory scale to a larger, posterior aligned scale. Let t denote the MCMC iteration. For the random walk proposal the standard deviation σ_{rw}^2 is defined as,

$$\sigma_{rw}(t) = 0.03 + 0.07\Big(\frac{min(t, 15000)}{15000}\Big). \tag{3}$$

This rule has the following three interpretation:

(i) **Early iterations (exploration):**

When t is less than 15000, the proposal scale is close to 0.03, producing small, conservative random walk steps that stabilize the chain while it adapts to the posterior geometry.

(ii) **Later iterations (posterior sensitive updates):**

As t increases, the proposal scale grows linearly to 0.10, encouraging larger moves and escaping local modes.

(iii) **Clipping at iteration 15000:**

This caps the adaptation after iteration 15000, preventing uncontrolled growth and maintaining a stable long run proposal distribution.

This adaptive random walk mechanism complements the spike and slab edge proposals by allowing flexible local exploration of the causal effect posterior, which is especially important for DCGs.

C Simulation Results Table

C.1 Acyclic Simulation Tables

		$p = 20$			$p = 30$			$p = 40$		
N	Metric	ICA-L	Direct-L	BayesDAG	ICA-L	Direct-L	BayesDAG	ICA-L	Direct-L	BayesDAG
	TPR	0.4246	0.8657	0.9584	0.3745	0.7347	0.9080	0.3161	0.6603	0.8161
	FPR	0.0407	0.0416	0.0201	0.0619	0.1020	0.0213	0.0713	0.1502	0.0239
	Acc	0.9189	0.9516	0.9778	0.8835	0.8821	0.9717	0.8688	0.8313	0.9604
70	Precision	0.4556	0.6612	0.8353	0.3922	0.4528	0.8229	0.3241	0.3256	0.7838
	Recall	0.4246	0.9000	0.9584	0.3745	0.7347	0.9080	0.3161	0.6603	0.8161
	F1	0.4359	0.7436	0.8917	0.3812	0.5576	0.8628	0.3188	0.4354	0.7988
	TPR	0.4133	0.9787	0.9710	0.4180	0.9267	0.9392	0.3689	0.8109	0.8939
	FPR	0.0304	0.0121	0.0158	0.0554	0.0396	0.0192	0.0693	0.1169	0.0211
	Acc	0.9277	0.9872	0.9829	0.8935	0.9571	0.9767	0.8757	0.8760	0.9704
100	Precision	0.5215	0.8845	0.8679	0.4469	0.7524	0.8449	0.3645	0.4445	0.8215
	Recall	0.4133	0.9787	0.9910	0.4180	0.9572	0.9392	0.3689	0.8109	0.8939
	F1	0.4582	0.9266	0.9158	0.4304	0.8252	0.8891	0.3657	0.5710	0.8558
	TPR	0.6810	1.0000	0.9595	0.3835	1.0000	0.9633	0.3992	0.9950	0.8818
	FPR	0.0243	0.0027	0.0125	0.0402	0.0087	0.0132	0.0496	0.0482	0.0288
	Acc	0.9632	0.9972	0.9849	0.9038	0.9923	0.9845	0.8965	0.9551	0.9622
200	Precision	0.8181	0.9682	0.8910	0.4605	0.9481	0.8944	0.4646	0.7150	0.7889
	Recall	0.6810	1.0000	0.9595	0.3835	0.9985	0.9633	0.3992	0.9849	0.8818
	F1	0.7399	0.9820	0.9234	0.4345	0.9654	0.9268	0.4286	0.8229	0.8312
	TPR	0.9905	1.0000	0.9797	0.4400	1.0000	0.9613	0.3451	0.9937	0.9665
	FPR	0.0042	0.0031	0.0110	0.0565	0.0062	0.0115	0.0506	0.0479	0.0115
	Acc	0.9953	0.9970	0.9899	0.8948	0.9933	0.9858	0.8909	0.9555	0.9863
300	Precision	0.9622	0.9619	0.9205	0.4605	0.9481	0.9055	0.4245	0.7243	0.9045
	Recall	0.9905	1.0000	0.9797	0.4400	0.9990	0.9613	0.3451	0.9937	0.9665
	F1	0.9699	0.9836	0.9488	0.4487	0.9712	0.9319	0.3798	0.8282	0.9339

Table 1: Performance comparison among three methods (ICALiNGAM, DirectLiNGAM, BayesDAG) in terms TPR, FPR, Acc, Precision, Recall, and F1: 100 simulated data sets on simulation setting; the error terms $\epsilon_i^{(q)} \sim \sum_{k=1}^{M_{true}} \pi_{ik} N(\mu_{ik}, \tau_{ik})$, where $M_{true} = 2$, $(\mu_{i1}, \mu_{i2}) = (-0.5, 0.5)$, and $(\tau_{i1}, \tau_{i2}) = (0.1, 0.3)$, and $(\pi_{i1}, \pi_{i2}) = (0.5, 0.5)$, $B = E$, $(\Delta, 1 - \Delta)$=(0.9,0.1); Average value of each metric is summarized in this table; We fit our methods with a misspecified mixture $M_{mis} = 5$.

		$p = 20$			$p = 30$			$p = 40$		
N	Metric	ICA-L	Direct-L	BayesDAG	ICA-L	Direct-L	BayesDAG	ICA-L	Direct-L	BayesDAG
	TPR	0.3927	0.7155	0.9362	0.3594	0.6668	0.8854	0.3016	0.6193	0.8128
	FPR	0.0359	0.0594	0.0196	0.0581	0.1028	0.0200	0.0670	0.1527	0.0200
	Acc	0.9208	0.9235	0.9762	0.8854	0.8750	0.9707	0.8712	0.8250	0.9638
70	Precision	0.4707	0.5206	0.8330	0.3994	0.4192	0.8276	0.3284	0.3080	0.8125
	Recall	0.3927	0.7155	0.9362	0.3594	0.6668	0.8854	0.3016	0.6193	0.8128
	F1	0.4237	0.5988	0.8806	0.3764	0.5130	0.8547	0.3131	0.4108	0.8119
	TPR	0.3983	0.8139	0.9557	0.3937	0.7599	0.9460	0.3679	0.7257	0.9207
	FPR	0.0274	0.0438	0.0163	0.0507	0.0814	0.0153	0.0658	0.1327	0.0145
	Acc	0.9294	0.9452	0.9810	0.8954	0.9032	0.9808	0.8788	0.8535	0.9791
100	Precision	0.5393	0.6296	0.8627	0.4541	0.5216	0.8700	0.3765	0.3812	0.8712
	Recall	0.3983	0.8139	0.9557	0.3937	0.7599	0.9460	0.3679	0.7257	0.9207
	F1	0.4550	0.7056	0.9061	0.4203	0.6152	0.9060	0.3711	0.4980	0.8949
	TPR	0.4803	0.9616	0.9480	0.3753	0.9789	0.9717	0.3900	0.9094	0.9456
	FPR	0.0320	0.0093	0.0145	0.0346	0.0282	0.0109	0.0466	0.0745	0.0143
	Acc	0.9313	0.9885	0.9820	0.9077	0.9685	0.9875	0.8984	0.9239	0.9816
200	Precision	0.5565	0.9058	0.8746	0.5359	0.8251	0.9063	0.4755	0.6054	0.8824
	Recall	0.4803	0.9616	0.9481	0.3753	0.9388	0.9717	0.3900	0.9094	0.9456
	F1	0.5124	0.9310	0.9091	0.4410	0.8726	0.9376	0.4277	0.7196	0.9125
	TPR	0.8930	0.9932	0.9779	0.4574	0.9903	0.9724	0.3280	0.9769	0.9578
	FPR	0.0220	0.0043	0.0105	0.0485	0.0084	0.0095	0.0479	0.0452	0.0124
	Acc	0.9714	0.9955	0.9883	0.9036	0.9914	0.9887	0.8910	0.9569	0.9854
300	Precision	0.7931	0.9555	0.9110	0.5059	0.9450	0.9181	0.4254	0.7376	0.8993
	Recall	0.8930	0.9932	0.9779	0.4574	0.9903	0.9724	0.3280	0.9930	0.9578
	F1	0.8371	0.9730	0.9271	0.4792	0.9655	0.9442	0.3696	0.8330	0.9593

Table 2: Performance comparison among three methods (ICALiNGAM, DirectLiNGAM, BayesDAG) in terms TPR, FPR, Acc, Precision, Recall, and F1: 100 simulated data sets on simulation setting; the error terms $\epsilon_i^{(q)} \sim Laplace(0, 0.25)$, $B = E$, $(\Delta, 1-\Delta)$=(0.9,0.1); Average value of each metric is summarized in this table; We fit our methods with a misspecified mixture $M_{mis} = 5$.

		$p = 20$			$p = 30$			$p = 40$		
N	Metric	ICA-L	Direct-L	BayesDAG	ICA-L	Direct-L	BayesDAG	ICA-L	Direct-L	BayesDAG
	TPR	0.3928	0.5960	0.9333	0.3535	0.5561	0.7652	0.3004	0.5557	0.6126
	FPR	0.0388	0.0829	0.0253	0.0590	0.1344	0.0440	0.0661	0.1662	0.0508
	Acc	0.9184	0.8932	0.9707	0.8835	0.8356	0.9371	0.8719	0.8065	0.9167
70	Precision	0.4507	0.3798	0.7977	0.3892	0.3128	0.6561	0.3301	0.2684	0.5614
	Recall	0.3928	0.5960	0.9333	0.3535	0.5561	0.7652	0.3004	0.5557	0.6126
	F1	0.4156	0.4615	0.8588	0.3686	0.3991	0.7058	0.3128	0.3613	0.5851
	TPR	0.3859	0.6383	0.9007	0.3920	0.5981	0.7725	0.3576	0.6022	0.6592
	FPR	0.0306	0.0777	0.0304	0.0524	0.1307	0.0501	0.0066	0.1703	0.0598
	Acc	0.9255	0.9010	0.9628	0.8953	0.8431	0.9324	0.8775	0.8076	0.9128
100	Precision	0.5039	0.4167	0.7634	0.4451	0.3363	0.6350	0.3690	0.2798	0.5452
	Recall	0.3859	0.6384	0.9007	0.3920	0.5981	0.7725	0.3576	0.6022	0.6592
	F1	0.4338	0.5011	0.8250	0.4156	0.4292	0.6960	0.3617	0.3813	0.5961
	TPR	0.4222	0.7456	0.8140	0.3680	0.7364	0.7693	0.3811	0.7057	0.6465
	FPR	0.0412	0.0554	0.0475	0.0377	0.0971	0.0641	0.0488	0.1592	0.0919
	Acc	0.9184	0.9295	0.9392	0.9044	0.8867	0.9196	0.8953	0.8277	0.8823
200	Precision	0.4617	0.5527	0.6679	0.5100	0.4701	0.5779	0.4583	0.3330	0.4432
	Recall	0.4222	0.7456	0.8140	0.4222	0.7364	0.7679	0.3680	0.7057	0.6465
	F1	0.4388	0.6301	0.7307	0.4261	0.5699	0.6574	0.4153	0.4507	0.5243
	TPR	0.6755	0.8011	0.8533	0.4279	0.8307	0.7674	0.3316	0.7735	0.7163
	FPR	0.0567	0.0437	0.0403	0.0559	0.0761	0.0727	0.0481	0.1386	0.0949
	Acc	0.9229	0.9468	0.9498	0.8941	0.9121	0.9116	0.8911	0.8528	0.8868
300	Precision	0.5048	0.6488	0.7073	0.4552	0.4279	0.5579	0.4277	0.3979	0.4617
	Recall	0.6755	0.8307	0.8533	0.8152	0.8011	0.7674	0.3316	0.7735	0.7163
	F1	0.5755	0.7217	0.7705	0.4399	0.6612	0.6426	0.3728	0.5211	0.5587

Table 3: Performance comparison among three methods (ICALiNGAM, DirectLiNGAM, BayesDAG) in terms TPR, FPR, Acc, Precision, Recall, and F1: 100 simulated data sets on simulation setting; the error terms $\epsilon_i^{(q)} \sim t_{df=7}$, $B = E$, $(\Delta, 1-\Delta)$=(0.9,0.1); Average value of each metric is summarized in this table; We fit our methods with a misspecified mixture $M_{mis} = 5$.

C.2 Cyclic Simulation Tables

		$p = 20$		$p = 30$		$p = 40$	
N	Metric	RGM	BayesDCG	RGM	BayesDCG	RGM	BayesDCG
	TPR	0.9955	0.3936	0.9947	0.3407	0.9961	0.2582
	FPR	0.9414	0.1044	0.9591	0.0842	0.9687	0.1301
	Acc	0.1452	0.8561	0.1289	0.8583	0.1214	0.8129
70	Precision	0.0973	0.3283	0.0954	0.2638	0.0957	0.1734
	Recall	0.9955	0.4751	0.9947	0.2969	0.9961	0.2582
	F1	0.1771	0.3848	0.1740	0.2767	0.1746	0.2046
	TPR	0.9935	0.6925	0.9964	0.6219	0.9958	0.3700
	FPR	0.9402	0.0756	0.9597	0.0667	0.9684	0.0897
	Acc	0.1462	0.9027	0.1286	0.9041	0.1216	0.8597
100	Precision	0.0973	0.5012	0.0955	0.4933	0.0957	0.3008
	Recall	0.9935	0.6925	0.9964	0.6219	0.9958	0.3700
	F1	0.1770	0.5784	0.1742	0.5487	0.1746	0.3303
	TPR	0.9880	0.9466	0.9949	0.9987	0.9949	1.0000
	FPR	0.9304	0.0298	0.9584	0.0125	0.9676	0.0189
	Acc	0.1546	0.9682	0.1296	0.9886	0.1222	0.9829
200	Precision	0.0977	0.7943	0.0955	0.8927	0.0957	0.8473
	Recall	0.9880	0.9466	0.9949	0.9987	0.9949	0.9999
	F1	0.1776	0.8579	0.1742	0.9424	0.1745	0.9170
	TPR	0.9724	0.9826	0.9901	0.9989	0.9945	1.0000
	FPR	0.9229	0.0165	0.9546	0.0090	0.9673	0.0134
	Acc	0.1599	0.9835	0.1326	0.9917	0.1225	0.9878
300	Precision	0.0970	0.8831	0.0954	0.9197	0.0957	0.8869
	Recall	0.9724	0.9826	0.9901	0.9989	0.9945	1.0000
	F1	0.1762	0.9257	0.1740	0.9574	0.1745	0.9398

Table 4: Performance comparison among two methods (RGM and BayesDCG) in terms TPR, FPR, Acc, Precision, Recall, and F1: 100 simulated data sets on simulation setting; the error terms $\epsilon_i^{(q)} \sim \sum_{k=1}^{M_{true}} \pi_{ik} N(\mu_{ik}, \tau_{ik})$, where $M_{true} = 2$, $(\mu_{i1}, \mu_{i2}) = (-0.5, 0.5)$, and $(\tau_{i1}, \tau_{i2}) = (0.1, 0.3)$, and $(\pi_{i1}, \pi_{i2}) = (0.5, 0.5)$, $(\Delta, 1-\Delta)$=(0.9,0.1); Average value of each metric is summarized in this table; We fit our methods with $M_{miss} = 5$.

		$p = 20$		$p = 30$		$p = 40$	
N	Metric	RGM	BayesDCG	RGM	BayesDCG	RGM	BayesDCG
	TPR	0.9954	0.5134	0.9963	0.2969	0.9970	0.1893
	FPR	0.9411	0.1121	0.9593	0.0842	0.9686	0.0670
	Acc	0.1455	0.8526	0.1289	0.8583	0.1215	0.8636
70	Precision	0.0974	0.3230	0.0955	0.2638	0.0958	0.2283
	Recall	0.9954	0.5134	0.9963	0.2969	0.9970	0.1893
	F1	0.1772	0.3936	0.1742	0.2767	0.1747	0.2045
	TPR	0.9939	0.6392	0.9970	0.4620	0.9954	0.2807
	FPR	0.9406	0.1156	0.9591	0.1077	0.9682	0.0867
	Acc	0.1459	0.8612	0.1292	0.8520	0.1217	0.8541
100	Precision	0.0973	0.3688	0.0956	0.3047	0.0957	0.2513
	Recall	0.9939	0.6271	0.9970	0.4620	0.9954	0.2807
	F1	0.1771	0.4651	0.1744	0.3645	0.1745	0.2628
	TPR	0.9879	0.8999	0.9955	0.8392	0.9954	0.6583
	FPR	0.9323	0.0787	0.9581	0.0863	0.9680	0.1059
	Acc	0.1528	0.9195	0.1299	0.9065	0.1219	0.8718
200	Precision	0.0975	0.5619	0.0956	0.5090	0.0957	0.3922
	Recall	0.9879	0.8999	0.9955	0.8392	0.9954	0.6583
	F1	0.1773	0.6874	0.1743	0.6319	0.1746	0.4900
	TPR	0.9796	0.9415	0.9917	0.9743	0.9938	0.8962
	FPR	0.9201	0.0578	0.9568	0.0523	0.9673	0.0896
	Acc	0.1631	0.9423	0.1308	0.9502	0.1224	0.9090
300	Precision	0.0979	0.6545	0.0954	0.6678	0.0956	0.5156
	Recall	0.9796	0.9415	0.9917	0.9743	0.9938	0.8962
	F1	0.1779	0.7670	0.1739	0.7902	0.1744	0.6533

Table 5: Performance comparison among two methods (RGM and BayesDCG) in terms TPR, FPR, Acc, Precision, Recall, and F1: 100 simulated data sets on simulation setting; the error terms $\epsilon_i^{(q)} \sim Laplace(0, 0.25)$, $(\Delta, 1-\Delta)$=(0.9,0.1); Average value of each metric is summarized in this table; We fit our methods with $M_{miss} = 5$.

		$p = 20$		$p = 30$		$p = 40$	
N	Metric	RGM	BayesDCG	RGM	BayesDCG	RGM	BayesDCG
	TPR	0.9949	0.6835	0.9965	0.4821	0.9938	0.4978
	FPR	0.9400	0.1157	0.9594	0.1633	0.9664	0.1757
	Acc	0.1465	0.8654	0.1288	0.8038	0.1233	0.7933
70	Precision	0.0974	0.4086	0.0956	0.2394	0.0957	0.2325
	Recall	0.9949	0.6835	0.9965	0.4821	0.9938	0.4978
	F1	0.1773	0.5062	0.1743	0.3175	0.1746	0.3155
	TPR	0.9952	0.7891	0.9952	0.5904	0.9966	0.4550
	FPR	0.9402	0.0928	0.9591	0.1486	0.9685	0.1880
	Acc	0.1464	0.8962	0.1290	0.8269	0.1215	0.7786
100	Precision	0.0974	0.4936	0.0955	0.3019	0.0958	0.2025
	Recall	0.9952	0.7891	0.9952	0.5904	0.9966	0.4550
	F1	0.1773	0.6027	0.1741	0.3966	0.1747	0.2789
	TPR	0.9878	0.9939	0.9938	0.8294	0.9913	0.6253
	FPR	0.9300	0.1019	0.9568	0.0949	0.9628	0.1593
	Acc	0.1549	0.9075	0.1310	0.8977	0.1262	0.8205
200	Precision	0.0977	0.7234	0.0956	0.4916	0.0958	0.2922
	Recall	0.9878	0.9939	0.9938	0.8294	0.9913	0.6253
	F1	0.1776	0.8108	0.1742	0.6144	0.1747	0.3972
	TPR	0.9715	0.9462	0.9894	0.9415	0.9902	0.7864
	FPR	0.9208	0.0490	0.9524	0.0610	0.9627	0.1745
	Acc	0.1617	0.9507	0.1346	0.9391	0.1263	0.8221
300	Precision	0.0972	0.6903	0.0956	0.6266	0.0957	0.3803
	Recall	0.9715	0.9462	0.9894	0.9415	0.9902	0.7864
	F1	0.1764	0.7933	0.1742	0.7501	0.1745	0.5031

Table 6: Performance comparison among two methods (RGM and BayesDCG) in terms TPR, FPR, Acc, Precision, Recall, and F1: 100 simulated data sets on simulation setting; the error terms $\epsilon_i^{(q)} \sim t_{df=7}$, $(\Delta, 1-\Delta)$=(0.9,0.1); Average value of each metric is summarized in this table; We fit our methods with $M_{miss} = 5$.