Post-selection inference for causal effects after causal discovery

Jian Yin, Chengli Zhang, Yuekai Li

City University of Hong Kong

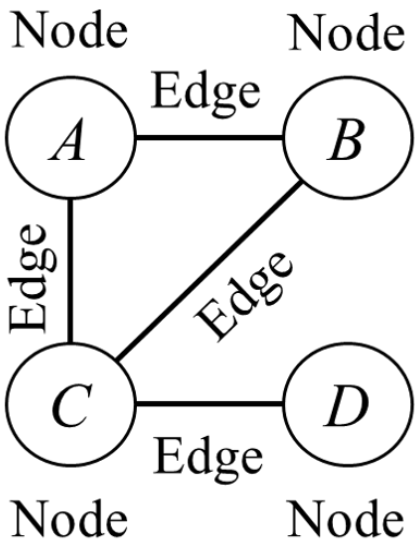
BIOS8005 Final Presentation

Contents

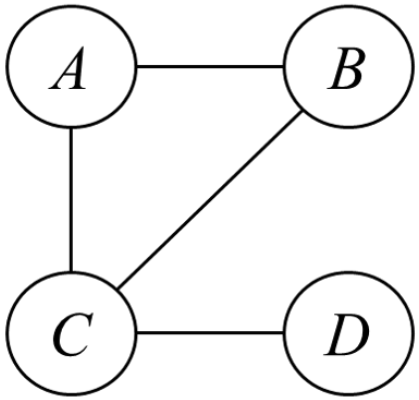
Introduction

Graph terminology

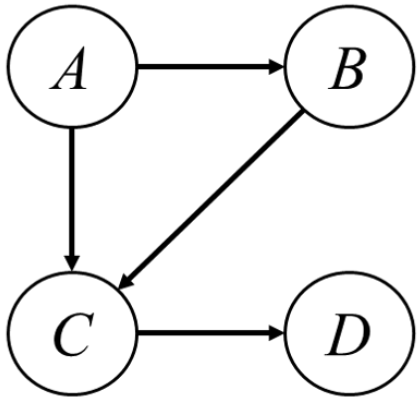
Nodes & Edges



Undirected Graph



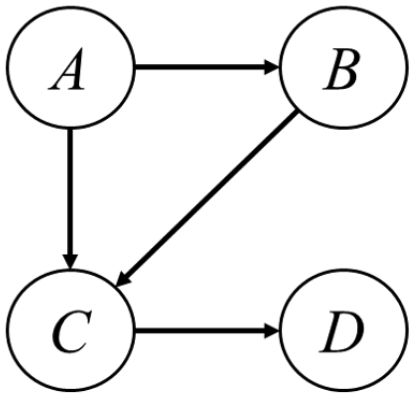
Directed Graph



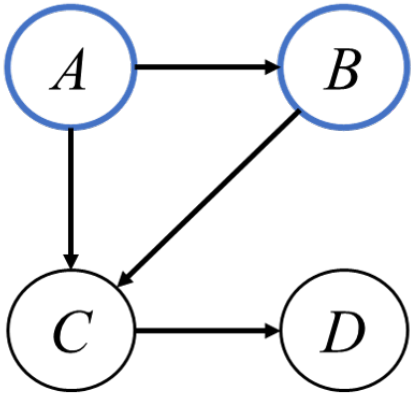
Graph terminology

Parent & Child

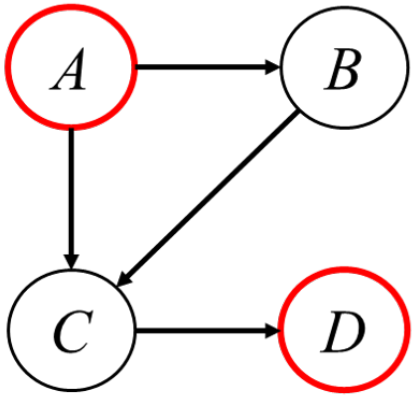
Parent Child



Adjacent

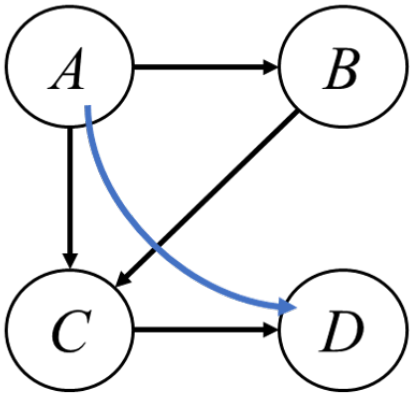


Not Adjacent

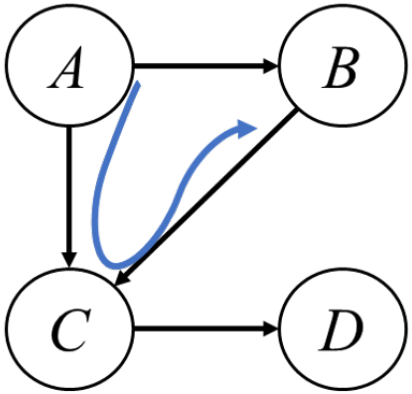


Graph terminology

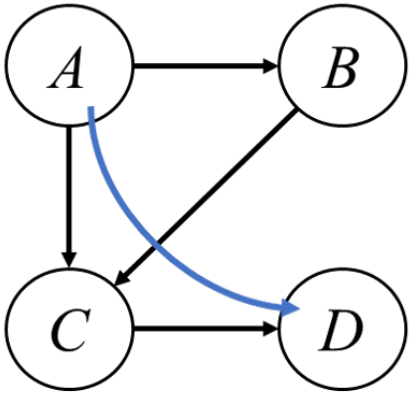
Path



Path

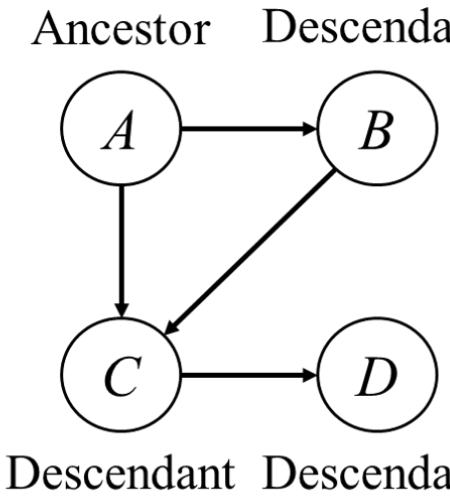


Directed Path

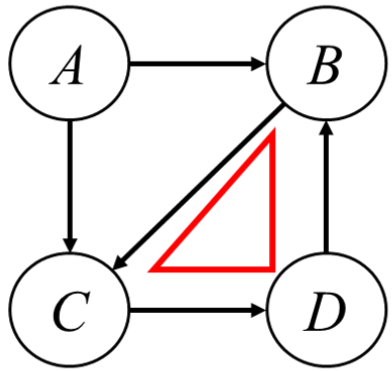


Graph terminology

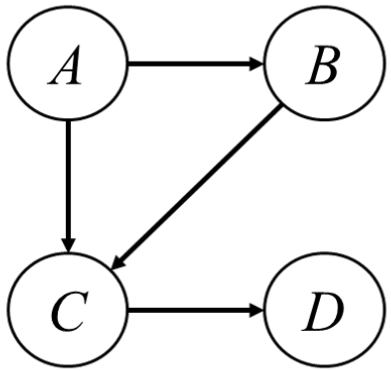
Ancestor & Descendant



Cycle



Directed Acyclic Graph (DAG)



Assumptions flowchart

Local Markov assumption

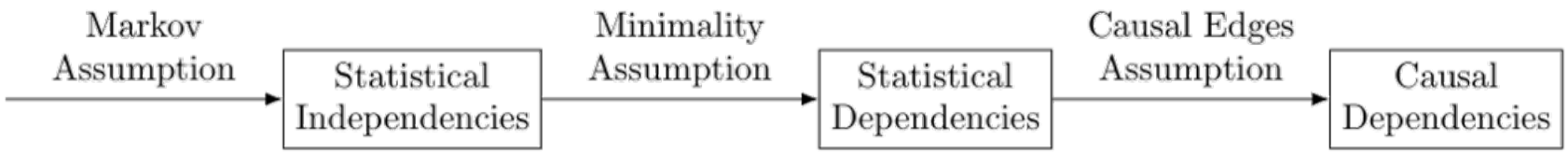
Given its parents in the DAG, a node X is independent of all of its non-descendants.

Minimality assumption

1. Local Markov assumption
2. Adjacent nodes in the DAG are dependent.

Causal edges assumption

In a directed graph, every parent is a direct cause of all its children.



Bayesian network factorization

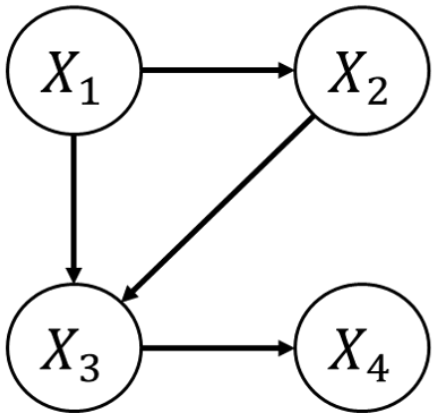
Bayesian network factorization

$$p(x_1, \ldots, x_p) = \prod_i p(x_i|\mathrm{pa}_i)$$

local Markov assumption $\Longrightarrow$ Bayesian network factorization

local Markov assumption $\Longleftarrow$ Bayesian network factorization

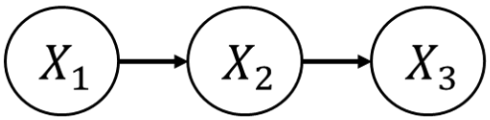
$$P(x_1, x_2, x_3, x_4)$$
$$= P(x_1)P(x_2|x_1)P(x_3|x_1, x_2)P(x_4|x_3)$$



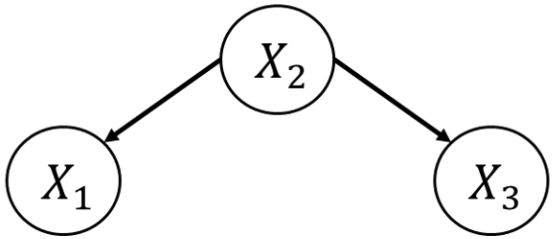
Graphical building blocks

Two nodes: X_1 X_2 or $X_1 \rightarrow X_2$

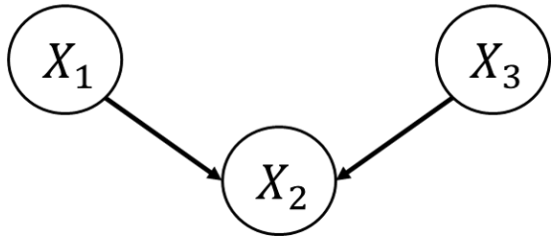
Chain



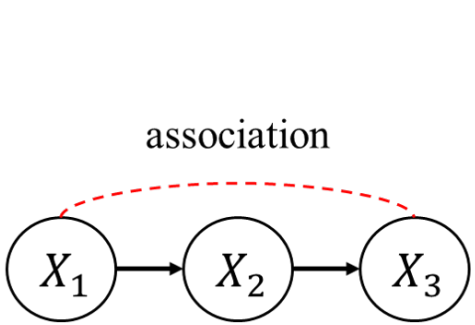
Fork



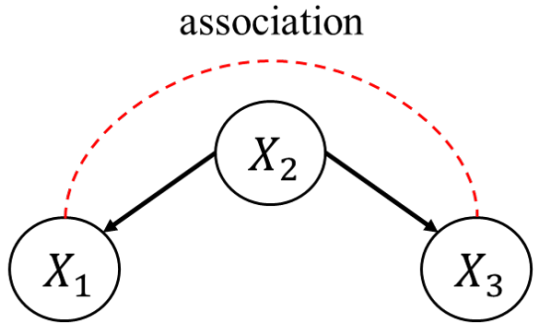
Immorality



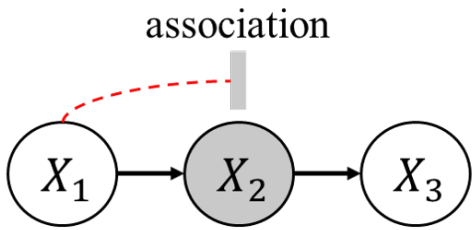
Chains and forks: dependence & independence



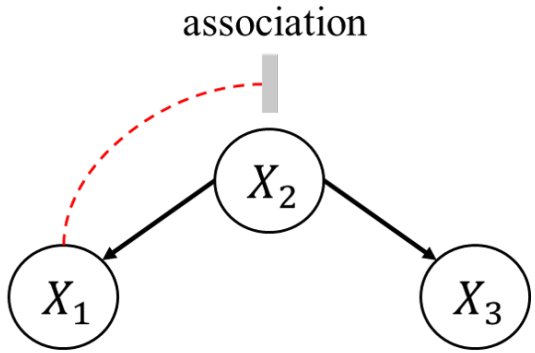
unblocked path



unblocked path



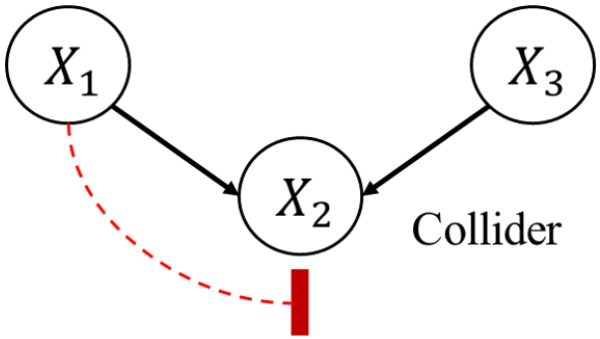
blocked path



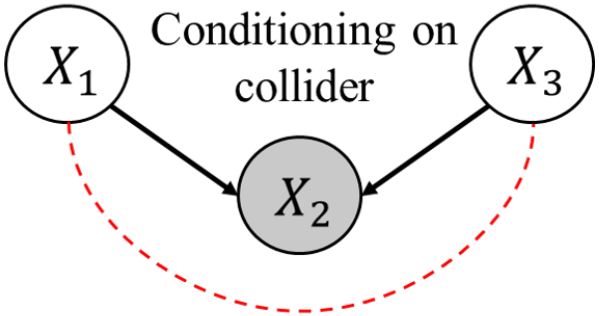
blocked path

Conditional Indepedence: $P(x_1, x_3|x_2) = P(x_1|x_2)P(x_3|x_2)$

Immoralities: dependence & independence



Blocked Path



Unblocked Path

$$
\begin{aligned}
P(x_1, x_3) &= \sum_{x_2} P(x_1, x_2, x_3) \\
&= \sum_{x_2} P(x_1)P(x_3)P(x_2|x_1, x_3) \\
&= P(x_1)P(x_3) \sum_{x_2} P(x_2|x_1, x_3) \\
&= P(x_1)P(x_3)
\end{aligned}
$$

d-separation

d-separation

Two (sets of) nodes X and Y are d-separated by a set of nodes Z if all of the paths between (any node in) X and (any node in) Y are blocked by Z.

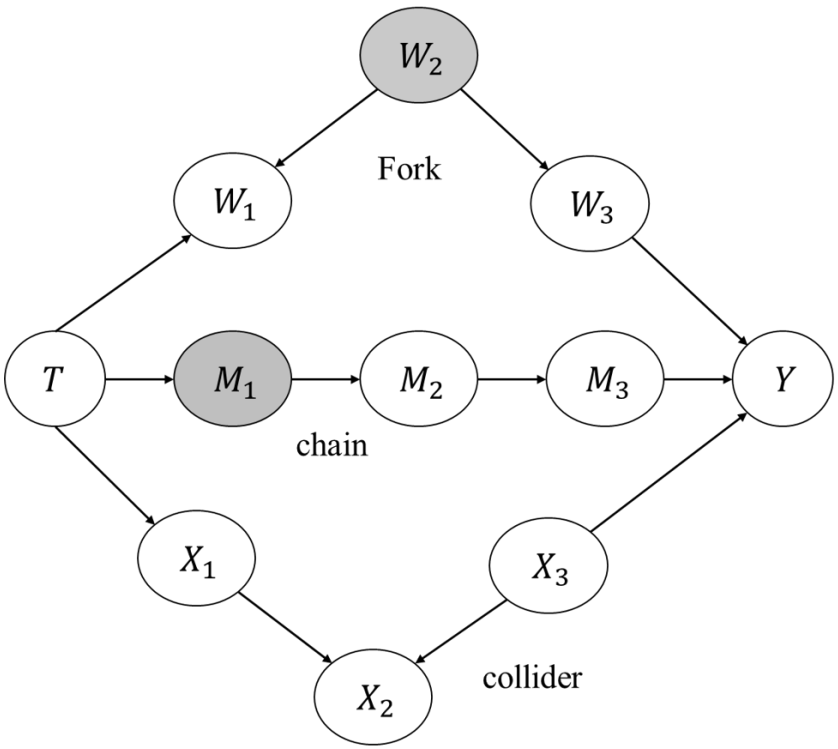
Theorem: Given that P is Markov with respect to G,

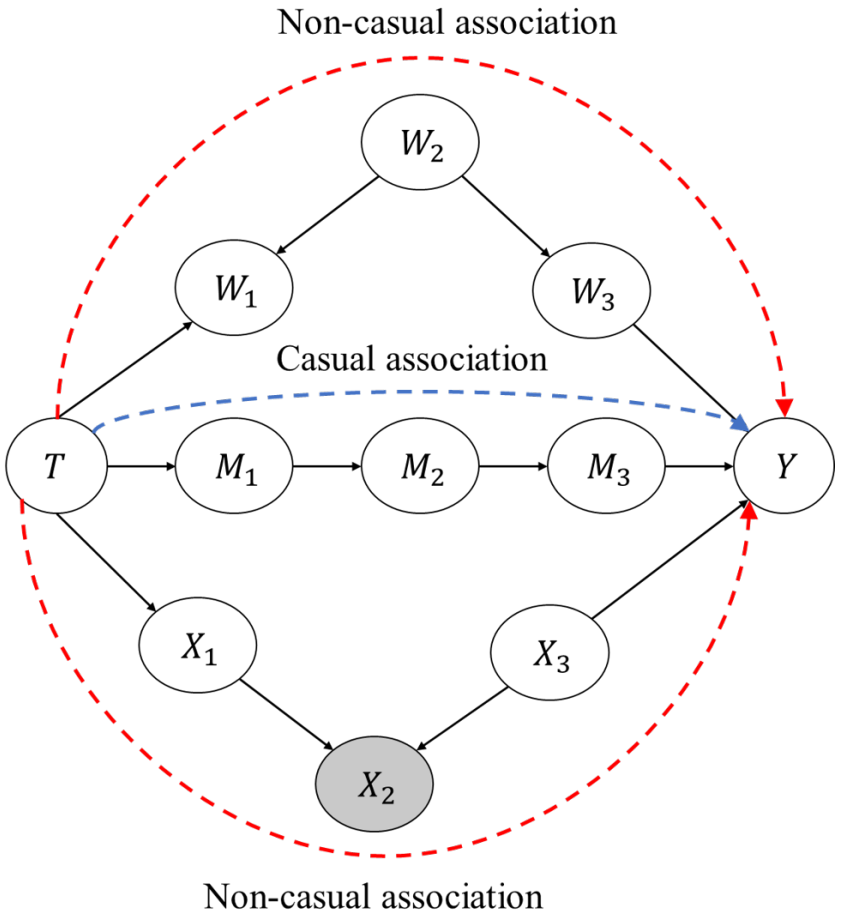
$$X \perp\!\!\!\perp_G Y \mid Z \implies X \perp\!\!\!\perp_P Y \mid Z$$

local Markov assumption $\iff$ global Markov assumption

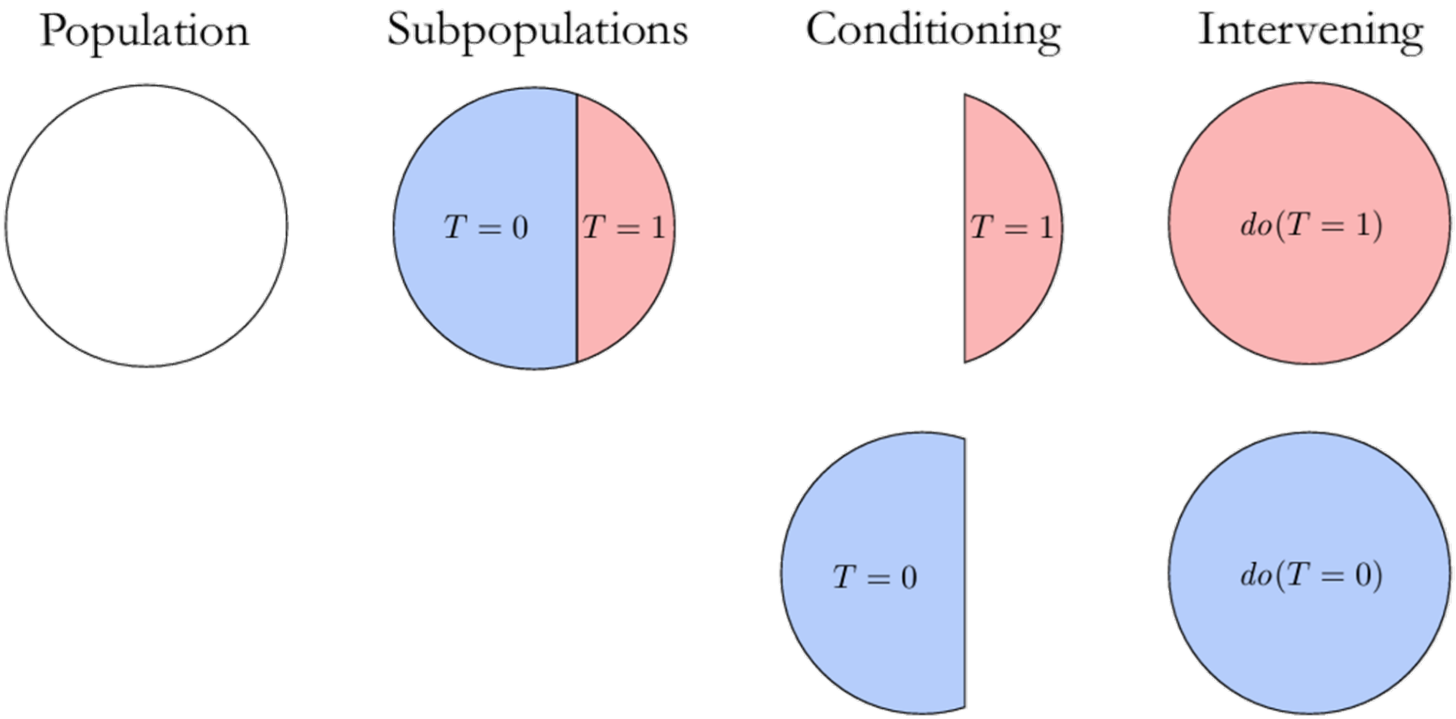
Markov assumption

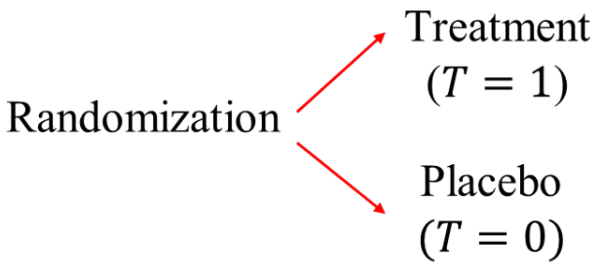
d-separation





Conditioning vs. intervening





do-operator

Interventional distributions:

$$P(Y(t) = y) \triangleq P(Y = y|do(T = t)) \triangleq P(y|do(t))$$

Average treatment effect (ATE):

$$\mathbb{E}[Y|do(T = 1)] - \mathbb{E}[Y|do(T = 0)]$$

Observational	Interventional
$P(Y, T, X)$	$P(Y|do(T = t))$
$P(Y|T = t)$	$P(Y|do(T = t), X = x)$

Truncated factorization

Bayesian network factorization

$$p(x_1, \ldots, x_p) = \prod_i p(x_i|\text{pa}_i)$$

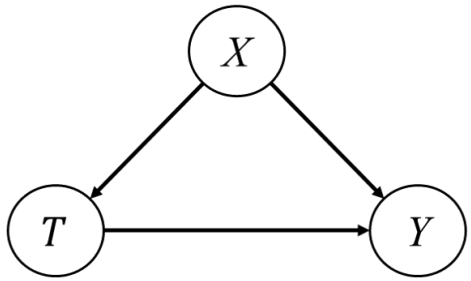
Truncated factorization

$$p(x_1, \ldots, x_p|do(S = s)) = \prod_{i \notin S} p(x_i|\text{pa}_i)$$

if x is consistent with the intervention.
Otherwise,

$$p(x_1, \ldots, x_p|do(S = s)) = 0$$

Truncated factorization



Bayesian network factorization: $P(y,t,x) = P(x)P(t|x)P(y|t,x)$

Truncated factorization: $P(y,x|do(t)) = P(x)P(y|t,x)$

Marginalize: $P(y|do(t),x) = \sum_x P(y|t,x)P(x)$

Backdoor criterion and backdoor adjustment

Backdoor criterion

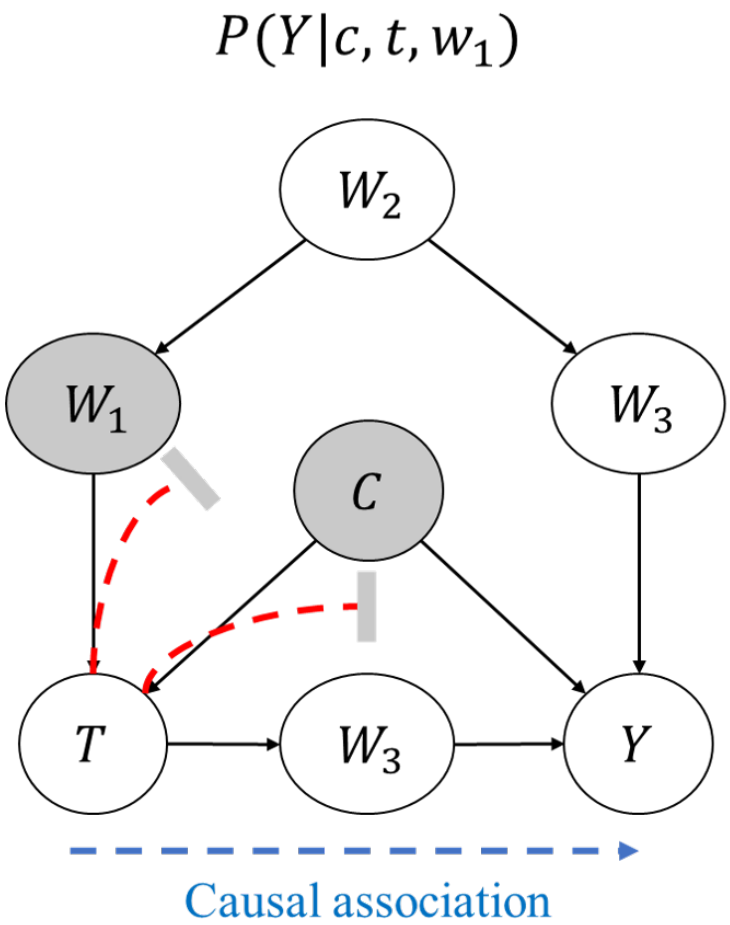
A set of variables W satisfies the backdoor criterion relative to T and Y if the following are true:

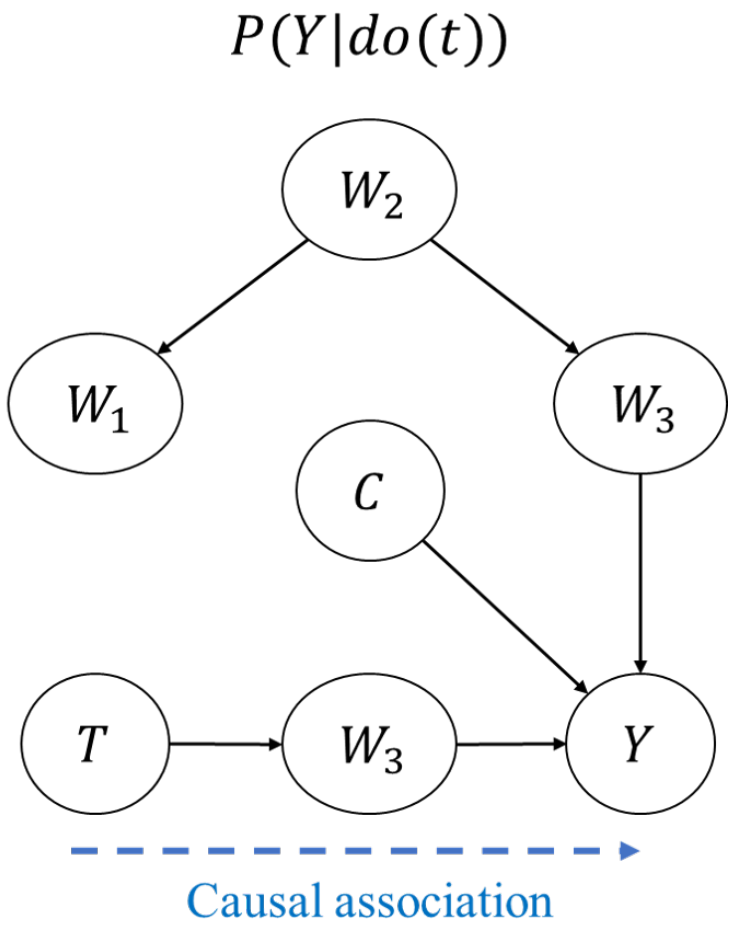
1. W blocks all backdoor paths from T to Y
2. W does not contain any descendants of T.

Given the modularity assumption and that W satisfies the backdoor criterion, we can identify the causal effect of X on Y:

$$P(Y|do(t)) = \sum_{w} P(y|t,w)P(w)$$

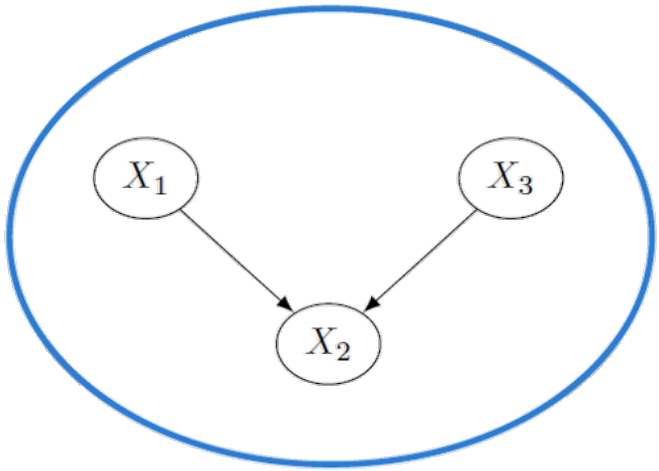
Backdoor criterion and backdoor adjustment



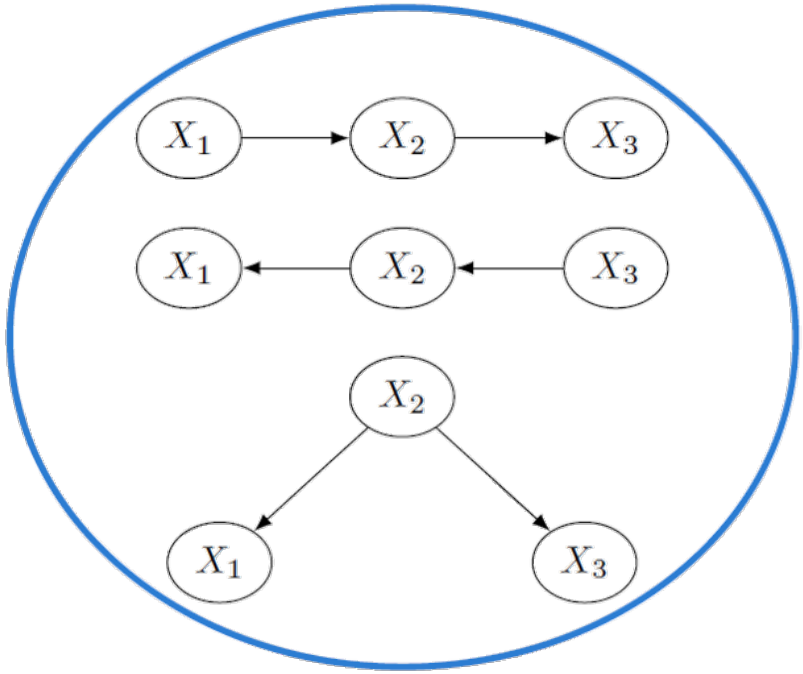


Markov Equivalence

Markov equivalence class where
$X_1 \perp\!\!\!\perp X_3$ and $X_1 \not\perp\!\!\!\perp X_3 \mid X_2$



Markov equivalence class where
$X_1 \perp\!\!\!\perp X_3 \mid X_2$ and $X_1 \not\perp\!\!\!\perp X_3$



Methodology

Problem Setup and Preliminaries — Outline

- **Notations & Setup**
  - DAGs, CPDAGs, causal effect definitions
  - Temporal ordering assumptions
- **Motivations**
  - Why post-selection inference matters
  - Uncertainty in graph selection
- **PC Algorithm (with temporal ordering)**
  - Adjacency search, v-structure detection
  - Time-tier constraints, Meek rules
- **Resampling-based Post-Selection Inference**
  - Two-step approach: Resampling + Aggregation
  - Screening invalid CPDAGs
- **Choice of Tuning Parameter**
  - Shrinkage parameter $\tau(M)$
  - Practical guidelines for c^*

Causal graphs and target of inference

Notation and Setup:

Let $G = (V, E)$ be the **true** DAG (Directed Acyclic Graph).

- $V = \{1, \ldots, d\}$: set of nodes (each node = a random variable X_i).
- $E \subset V \times V$: set of directed edges (direct causal relations).

For a node i:

- $\mathrm{Adj}_i(G)$: adjacency set (all nodes connected to i by an edge).
- $\mathrm{Pa}_i(G)$: parent set (direct causes of i).

A CPDAG C represents the Markov equivalence class of G:

- Mixed graph (directed and undirected edges).
- A directed edge $i \to j$ in C means $i \to j$ is in all DAGs Markov equivalent to G.

Target Causal Effect:

- We focus on the average treatment effect $\beta_{i,j}(G)$ of exposure i on outcome j.
- Using Pearl's do-notation:

$$\beta_{i,j}(G) \;=\; E\big[X_j \,\big|\, \mathrm{do}(X_i = x_i)\big] \;-\; E\big[X_j \,\big|\, \mathrm{do}(X_i = x_i')\big].$$

- Note: We aim for the *true* causal effect, not just a coefficient from one selected graph.

Why Post-Selection Inference?

Motivation:

- If we run the PC algorithm only *once*, using the *same data* for both model selection and inference, we end up treating the resulting graph as fixed.
- This ignores *model-selection uncertainty* and can lead to *overly optimistic* confidence intervals, because we fail to account for the random variation in graph structure.
- In other words, standard inference procedures would wrongly assume the selected model is the "true" one (a form of data reuse bias).

Our Strategy:

- We *resample* the test statistics (akin to a parametric bootstrap), then rerun the PC algorithm M times.
- Each run may yield a different CPDAG, capturing the randomness in the selection procedure.
- We discard any invalid graphs (cycles, bi-directed edges), and *aggregate* the remaining ones by taking the union of confidence intervals.

Methodology

Temporal Ordering:

- In many real-world problems, we know (or suspect) certain variables occur earlier in time than others.
- $\mathrm{O}(V)$ is a vector specifying partial temporal order (or tiers).
- E.g., $\mathrm{O}(V) = (1, 1, 2)$ for a 3-node system indicates variables 1,2 precede variable 3.

Benefits of Using Temporal Ordering:

- Avoid testing irrelevant conditioning sets (if a variable is in a strictly later tier than both i and j).
- Prevent orientations contradicting known time flow (no edges from future variables back to past).
- In practice: reduces false positives, speeds up PC search.

Methodology

(1) Conditional Independence Restriction:

- $\mathrm{CI}(X_i, X_j \mid X_S)$ is *not* tested if any variable in S is later in time than both i and j.
- Rationale: Temporally later nodes can't be valid conditioning sets for earlier exposures.

(2) Orientation Constraint:

- When finalizing edge directions, we disallow edges going from a later tier to an earlier tier (i.e., no backward edges in time).
- Ensures edges respect $\mathrm{O}(V)$.

$\overline{\mathrm{O}}_{ij}$ and $\mathrm{O}(V)$ Definitions

$\mathrm{O}(V)$: **partial temporal order vector**

- For each node $v \in V$, $\mathrm{O}(v) =$ "tier" or "time point" of v.
- If $\mathrm{O}(i) < \mathrm{O}(j)$, then node i is strictly earlier than j.
- If all variables have the same tier, no extra constraint is imposed.

$\overline{\mathrm{O}}_{ij}$:

- The set of nodes that are in strictly later tiers than both i and j.
- Example: If $\mathrm{O}(V) = (1, 1, 2)$ for a 3-node graph, then $\overline{\mathrm{O}}_{12} = \{3\}$.
- This set is used to skip certain conditional independence tests where future nodes would be invalid conditioning variables.

Fisher z-Transformation — Why and How

Partial Correlation in Gaussian Case:

- If X is multivariate Gaussian, $X_i \perp X_j \mid X_S \iff \rho_{ij|S} = 0$.
- $\rho_{ij|S}$ can be estimated from sample partial correlations.

Fisher z-Transformation:

$$Z(\rho_{ij|S}, n) = \sqrt{n - |S| - 3}\, \ln\Big(\frac{1 + \rho_{ij|S}}{1 - \rho_{ij|S}}\Big).$$

- Variance stabilization: helps each test statistic share a standard normal distribution under H_0.
- Used in the PC procedure to decide whether to remove an edge at a given α.

PC Algorithm (Simplified)

Algorithm 1: PC(Test, α, $\mathrm{O}(V)$) with Temporal Ordering

1. Form a complete graph $\tilde{C}$ on node set V.
2. Let $s = 0$.
3. **repeat**
   - 3.1 For each node i, define $\mathrm{Adj}_i(\tilde{C})$.
   - 3.2 For each pair (i, j) still adjacent:
     - If $\left|\tilde{\mathrm{Adj}}_i(\tilde{C}) \setminus \{j, \overline{\mathrm{O}}_{ij}\}\right| \geq s$,
     - Then test subsets $S \subseteq \tilde{\mathrm{Adj}}_i(\tilde{C}) \setminus \{j, \overline{\mathrm{O}}_{ij}\}$ with $|S| = s$.
     - If $X_i \perp X_j \mid X_S$, remove edge $i - j$ and record S in $\mathrm{sepset}_{i,j}$.
   - 3.3 Let $s = s + 1$.
4. **until** no more edges can be removed for the next size s.
5. (a) Determine v-structures, respecting $\mathrm{O}(V)$.
   (b) Apply Meek rules, respecting $\mathrm{O}(V)$.
6. Return CPDAG C.

PC Algorithm (cont'd)

(a) V-structures (Colliders)

For all triples (i, k, j) such that $i \in \mathrm{Adj}_k(\tilde{C})$ and $j \in \mathrm{Adj}_k(\tilde{C})$ but $i \notin \mathrm{Adj}_j(\tilde{C})$, orient $i - k - j$ as $i \to k \leftarrow j$ (called a v-structure or collider) if and only if $k \notin \mathrm{sepset}_{i,j}$.

(b) Meek rules

- Each rule leverages already oriented edges (or collider findings) to imply new directions, ensuring no directed cycles arise.
- Simplified intuition:
  - If orienting an edge one way would create a cycle or contradict a known collider structure, it must be oriented the opposite way.
  - Respect temporal ordering: disallow edges from future-tier variables to earlier-tier ones.
- Applied exhaustively until no further orientations can be deduced.

Methodology: Step 1

Resample Test Statistic:

- In the multivariate Gaussian case, partial correlation test uses

$$Z\big(\hat{\rho}_{ij|S}, n\big) = \sqrt{n - |S| - 3}\, \ln\Big(\frac{1 + \hat{\rho}_{ij|S}}{1 - \hat{\rho}_{ij|S}}\Big).$$

- Resampling: For each run $m = 1, \ldots, M$, draw

$$Z\big(\hat{\rho}^{[m]}_{ij|S}, n\big) \overset{\text{i.i.d.}}{\sim} N\Big(Z\big(\hat{\rho}_{ij|S}, n\big),\, 1\Big).$$

Rejection Region:

- We reject $H_0 : \rho_{ij|S} = 0$ if

$$\big|Z(\hat{\rho}^{[m]}_{ij|S}, n)\big| \;>\; \tau(M) \;\cdot\; z_{\nu/(2L)}.$$

- $\tau(M) \in (0, 1)$ is a shrinkage parameter, $z_{\nu/(2L)}$ is a normal critical value, and L upper-bounds the total # of tests in a single PC run.

Methodology: Step 1

Algorithm 2: Executing the PC-Algorithm M times

- **Input:** Samples of $X = (X_1, \ldots, X_d)'$ and $\mathrm{O}(V)$.
- **Output:** A collection of M graphs $\hat{C}^{[m]}$.

Algorithm 2: Resampling and Screening

---

for $m = 1, \ldots, M$ **do**

1. Run PC-Algorithm 1, but each CI test uses resampled statistic $Z(\hat{\rho}^{[m]}_{ij|S}, n)$.
2. Return the resulting CPDAG, call it $\hat{C}^{[m]}$.

end for

return $(\hat{C}^{[1]}, \ldots, \hat{C}^{[M]})$.

Discarding Invalid CPDAGs:

- Among the M outputs $\hat{C}^{[m]}$, some may contain cycles or bi-directed edges.
- Such graphs are deemed *invalid*, as they do not correspond to any DAG.
- Keep only

$$\mathcal{M} = \left\{ 1 \leq m \leq M : \hat{C}^{[m]} \text{ is a valid CPDAG} \right\}.$$

Methodology: Step 2

Motivation:

- A valid CPDAG $\widehat{C}^{[m]}$ may represent multiple DAGs (same Markov equivalence class).
- For each DAG, we can estimate $\beta(G)$ by back-door adjustment.
- Combine these estimates within each CPDAG to reflect uncertainty in directions of undirected edges.

Procedure (within one CPDAG $\widehat{C}^{[m]}$):

- Let k_m be the number of DAGs in $\widehat{C}^{[m]}$.
- For each DAG $\widehat{G}_k^{[m]}$,

$$\hat{\beta}_k^{[m]} \;=\; \hat{\beta}\big(\widehat{G}_k^{[m]}\big), \quad 1 \le k \le k_m.$$

- Use back-door adjustment (e.g., treat $\mathrm{Pa}_i(\widehat{G}_k^{[m]})$ as confounders) to compute estimates and standard errors.
- Construct

$$\mathrm{CI}^{[m]} \;=\; \bigcup_{k=1}^{k_m} \big(\hat{\beta}_k^{[m]} \;\pm\; z_{\alpha_1/2}\, \hat{\sigma}_{\beta_k}^{[m]} \big).$$

Methodology: Step 2

Union Across Multiple Runs:

- Recall:
$$\mathcal{M} = \{1 \leq m \leq M : \hat{C}^{[m]} \text{ is a valid CPDAG}\}.$$
- We already formed each $\mathrm{CI}^{[m]}$ for $m \in \mathcal{M}$.
- The final $(1-\gamma)\%$ confidence interval for $\beta(G)$ is
$$\mathrm{CI}^{\mathrm{re}} = \bigcup_{m \in \mathcal{M}} \mathrm{CI}^{[m]}.$$

Interpretation and Remarks:

- Ensures coverage if at least one $\widehat{C}^{[m]}$ is close to the true graph.
- May be conservative (taking a union can widen intervals).
- In practice, other more efficient adjustments or DAG expansions (beyond naive back-door) are possible.

Shrinkage Parameter $\tau(M)$

Under certain conditions (see paper for details), we can set:

$$\tau(M) = c^* \left(\frac{\log n}{M}\right)^{\frac{1}{L}}, \quad \text{where } c^* \text{ is a positive constant.}$$

- This choice satisfies the requirements for the resampling threshold in our method.
- L is the user-specified upper bound on the total number of conditional independence tests.
- As M grows, $\tau(M)$ shrinks accordingly, improving the chance of recovering the true structure *in at least one resample*.

Practical Choice of c^*

Why c^* matters:

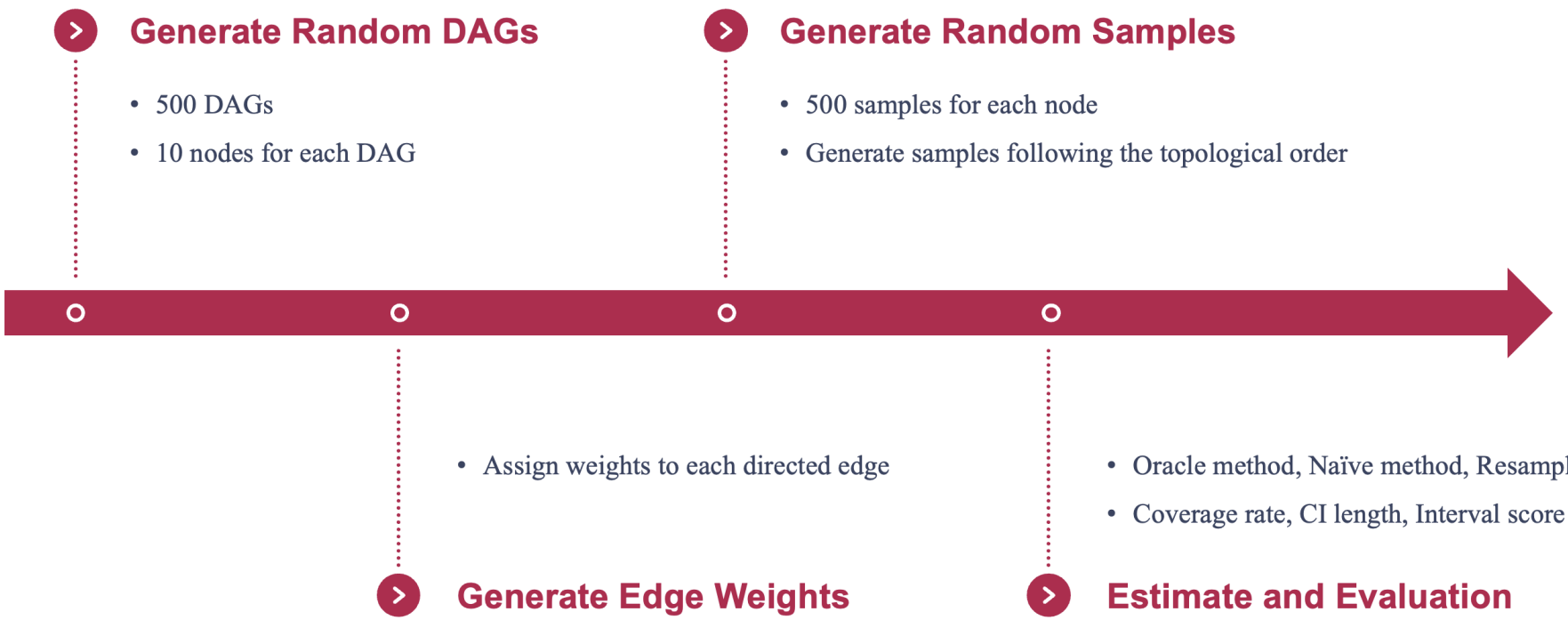
- $\tau(M) = c^* \left(\frac{\log n}{M}\right)^{1/L}$ is crucial in controlling edge-removal threshold.
- Larger $c^* \Rightarrow$ bigger threshold $\Rightarrow$ sparser graph (might lose important edges).
- Smaller $c^* \Rightarrow$ smaller threshold $\Rightarrow$ denser graph (higher risk of invalid CPDAGs or cycles).

Heuristic for choosing c^*:

- *Percentage of kept graphs* $\left|\mathcal{M}\right|/M$:
  - If too few CPDAGs survive (very small $|\mathcal{M}|$), coverage might degrade.
  - If too many survive (very large $|\mathcal{M}|$), the final union of intervals may become overly broad.
- Simulation-based approach:
  - Gradually increase c^* until the fraction $\left|\mathcal{M}\right|/M$ *is minimized* or falls below a target.
  - Check coverage or interval widths in a pilot run.

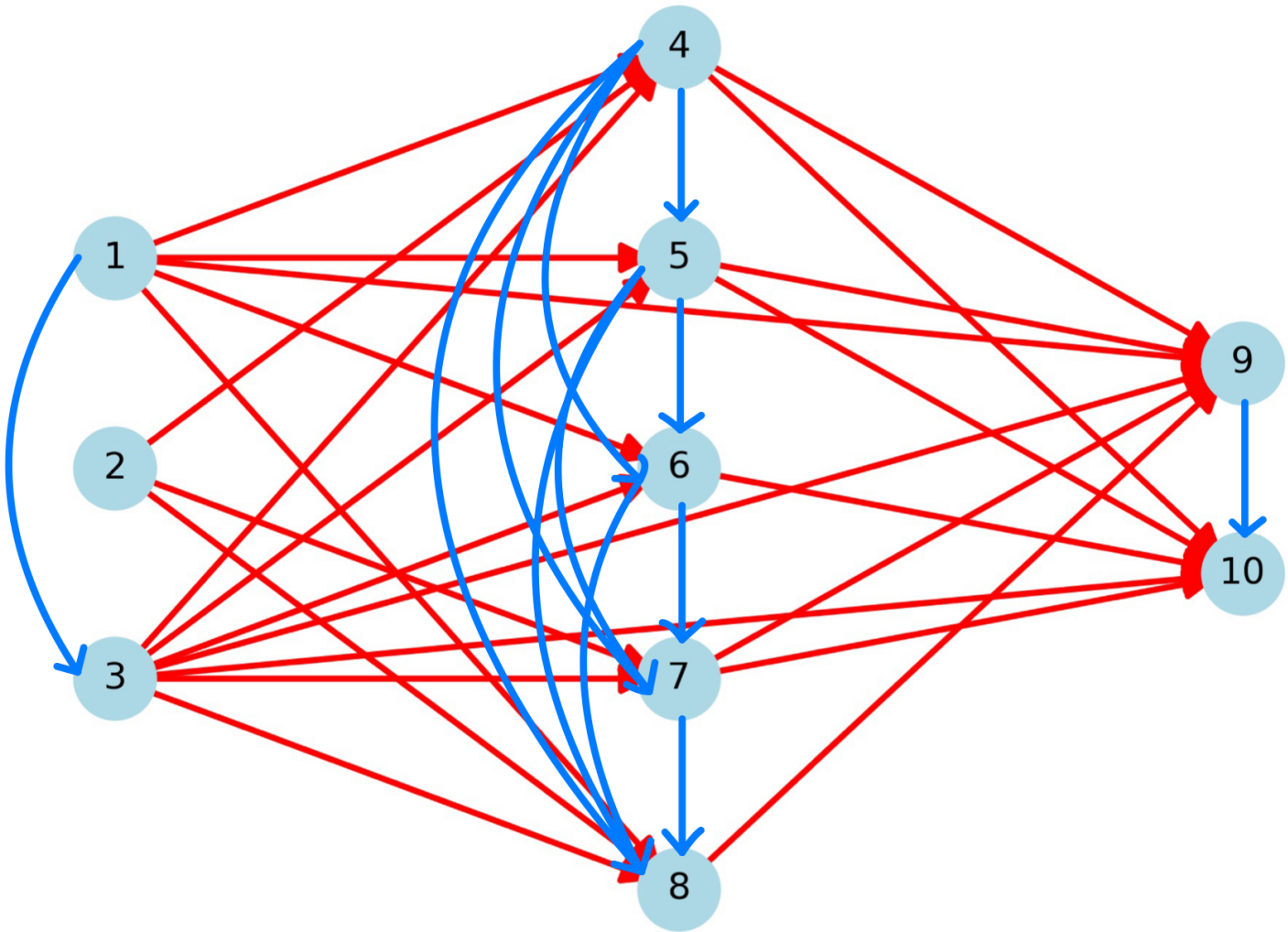
Simulations

Workflow


Generate Random DAGs
• 500 DAGs
• 10 nodes for each DAG
Generate Random Samples
• 500 samples for each node
• Generate samples following the topological order
• Assign weights to each directed edge
Generate Edge Weights
• Oracle method, Naïve method, Resample method
• Coverage rate, CI length, Interval score
Estimate and Evaluation

Data generation: Random DAGs

- We simulate 10-node random DAGs with a high density, and the expected sum of in-and-out degrees is 7 per node. For example:



Data generation: Random DAGs

Method

- We fix the node ordering as $1, 2 \ldots, 10$.
- Only directed edge $i \to j$ with $i < j$ are allowed, ensuring acyclicity.
- Out of the possible 45 edges from complete DAG, 35 are randomly selected.

Discussion

- The simulated DAGs are topologically ordered, so j is a non-ancestor of i.
- As long as partial temporal order is also arranged from small to large, the simulated DAGs will not violate the temporal order.
- We can assume partial knowledge of the temporal ordering of variables:

$$\mathrm{O}(V) = (1, 1, 1, 2, 2, 2, 2, 2, 3, 3)$$

Data generation: Edge Weights

Edge weight generation

- For each selected directed edge $i \to j$, a preliminary weight $\widetilde{w}_{ij}$ is drawn uniformly from $(-1, -0.5) \cup (0.5, 1)$.
- Then, the weights are scaled, and the final weight is computed as:

$$w_{ij} = \frac{\widetilde{w}_{ij}}{\sqrt{1 + \sum_{k \in \mathsf{Pa}_j(G)} \widetilde{w}_{kj}^2}} \text{ for } i, j = 1, \ldots, 10$$

- This scaling ensures that if all parent variables are $\mathcal{N}(0, 1)$, then X_j would have unit variance.

Data generation: Random Samples

Simulating the Data

- The data vector $X = (X_1, \ldots, X_{10})$ is generated following the topological order.
- For each node $j = 1, \ldots, 10$, the value is given by:

$$X_j = \sum_{k \in \mathsf{Pa}_j(G)} w_{kj} X_k + \varepsilon_j, \quad \varepsilon_j \sim \mathcal{N}(0, 1)$$

- This procedure is applied sequentially starting from the first node.

Target estimand

- Our target estimand is the effect of variable 6 on variable 10:

$$\beta_{6,10}(G) = w_{6,10}$$

Methods for Comparison

Oracle method

- Oracle method:

$$\text{CI}^{oracle} = \hat{\beta}(G) \pm 1.96\hat{\sigma}_{\beta}^{oracle}$$

Naive method

- Naive method:

$$\text{CI}^{naive} = \bigcup_{k} \hat{\beta}_k(\hat{G}_k) \pm 1.96\hat{\sigma}_{\beta_k}^{naive}$$

Resample method

- Resample method:

$$\text{CI}^{resample} = \bigcup_{m \in \mathcal{M}} \bigcup_{k} \hat{\beta}_k^{[m]} \pm z_{(\gamma-\nu)/2}\hat{\sigma}_{\beta_k}^{[m]}$$

Evaluation Metrics

Coverage Rate

The coverage rate is the proportion of times the true parameter is contained within the confidence interval.

CI Length

The confidence interval (CI) length is defined as the difference between the upper and lower bounds:

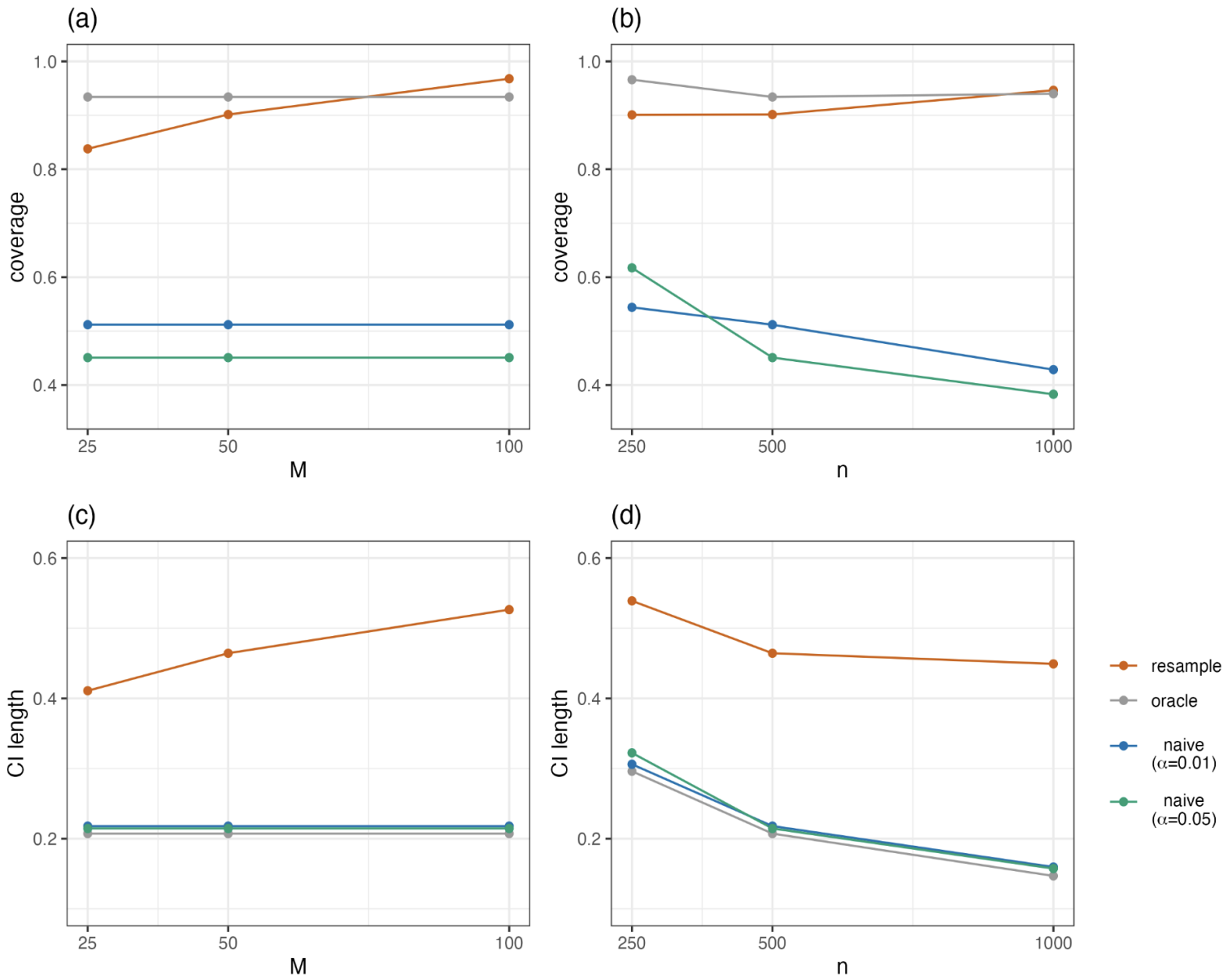
$$\text{CI Length} = U - L.$$

Interval Score

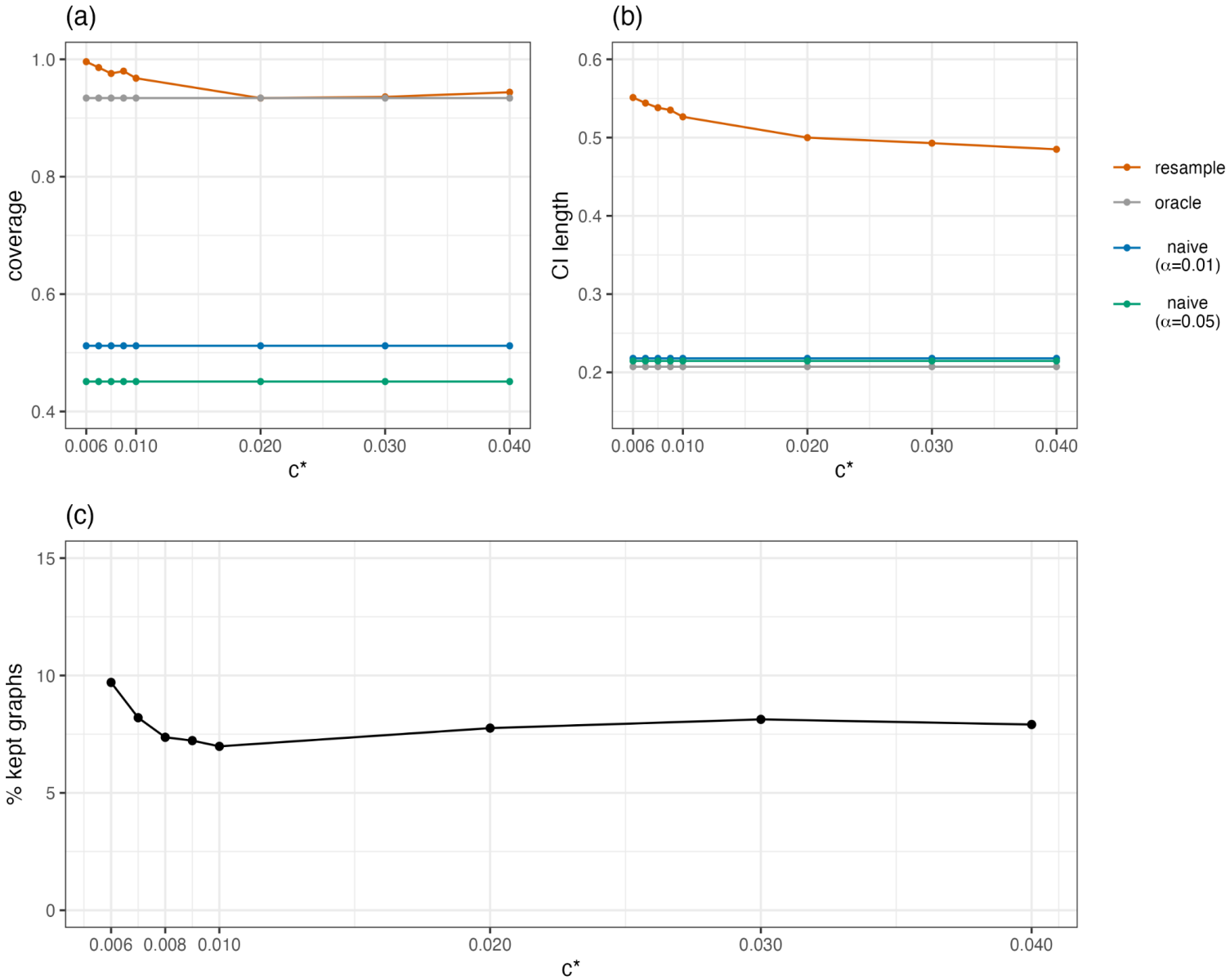
It is designed to consider both the coverage and length. For a given confidence level $1 - \alpha$, and the true value y.

$$S_\alpha(L, U; y) = (U - L) + \frac{2}{\alpha}(L - y)\mathbf{1}\{y < L\} + \frac{2}{\alpha}(y - U)\mathbf{1}\{y > U\}.$$

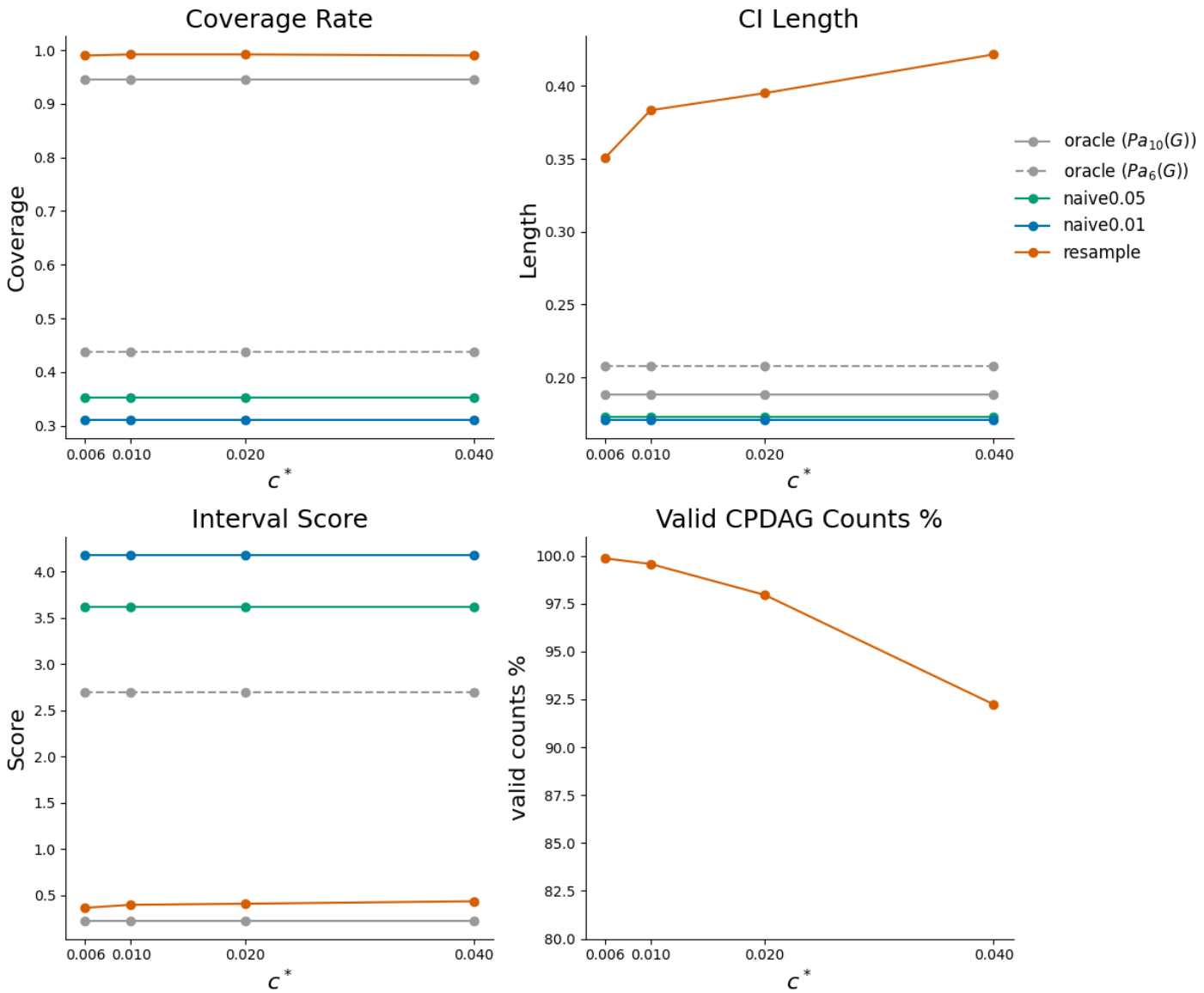
Results from the paper



Results from the paper (cont'd)



Results reproduced by our code*



*Github repository: `https://github.com/DuckLeeyk/Post-Selection-PC-Algorithm`