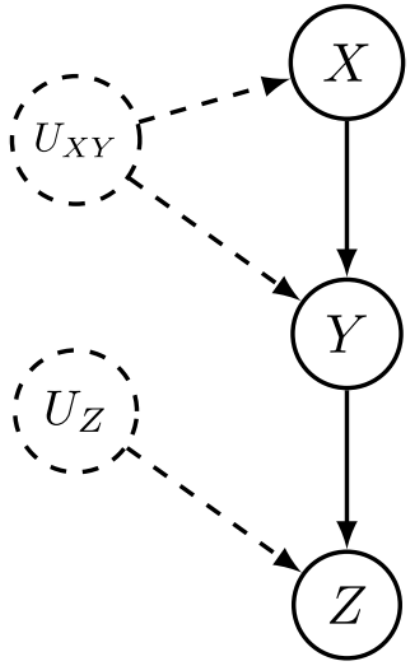
# Week 12

Wentao Gao

# Causal structure learning

1. Structure causal models
2. Some essential concepts
3. Methods for Causal Structure Learning

# Structural Causal Models




(a)

$$M = (\mathbf{V}, \mathbf{U}, \mathbf{F}, P)$$
$$\mathbf{V} = \{X, Y, Z\}$$
$$\mathbf{U} = \{U_{XY}, U_Z\}$$
$$\mathbf{F} = \begin{cases} f_X : X := 2U_{XY}, \\ f_Y : Y := X + U_{XY}, \\ f_Z : Z := 3Y + U_Z \end{cases}$$
$$P = \begin{cases} U_{XY} \sim \mathcal{N}(0, 1), \\ U_Z \sim \mathcal{N}(0, 1) \end{cases}$$

(b)

**Definition 2.10** *(Structural causal model).* A *structural causal model* (SCM) [9,18] is defined by the tuple $M = (\mathbf{V}, \mathbf{U}, \mathbf{F}, P)$, where:

- $\mathbf{V}$ is a set of *endogenous* variables, i.e. *observable* variables,
- $\mathbf{U}$ is a set of *exogenous* variables, i.e. *unobservable*[1] variables, where $\mathbf{V} \cap \mathbf{U} = \varnothing$,
- $\mathbf{F}$ is a set of *functions*, where each function $f \in \mathbf{F}$ is defined as $f_i : (\mathbf{V} \cup \mathbf{U})^p \to \mathbf{V}$, with $p$ the ariety of $f$, so that $f$ determines completely the value of $V_i$,
- $P$ is a joint probability distribution over the exogenous variables $P(\mathbf{U}) = \prod_i P(U_i)$.

# Causal discovery problem

**The causal discovery problem consists in recovering the true graph $G^*$ from the given dataset D.**

A causal discovery algorithm is said to solve the causal discovery problem if and only if it converges to the true graph $G^*$ in the limit of the sample size of the dataset D

## Consistency of a causal graph

A causal discovery algorithm is consistent if it outputs a graph G that induces a probability distribution consistent with the input dataset D

## Identifiability of a causal graph

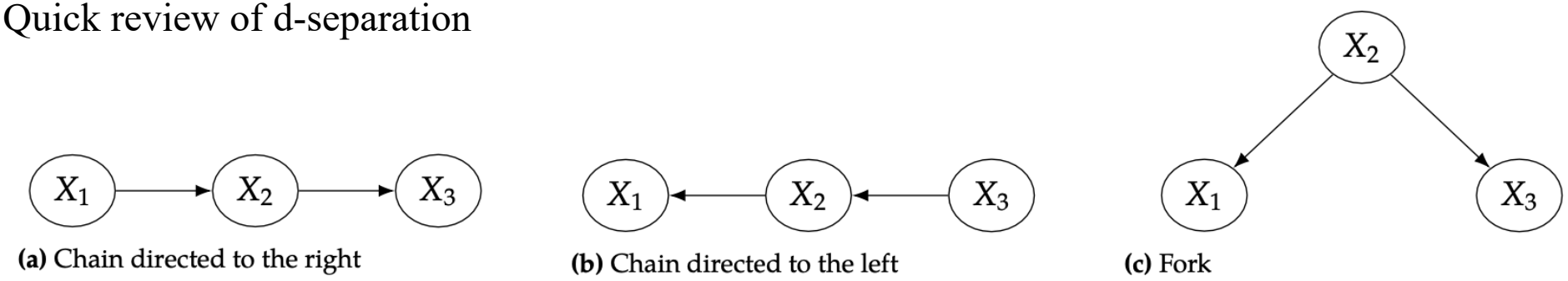
Causal discovery algorithm is said to identify a graph G if it is able to determine the direction of any edge in G.

# Markov properties and Markov equivalence in DAGs

**Markov property:** A graph G = (V, E) is said to satisfy the Markov property if the associated joint probability distribution P(V) can be decomposed recursively as:

$$P(\mathbf{V}) = \prod_{X \in \mathbf{V}} P(X|Pa(X))$$

Quick review of d-separation

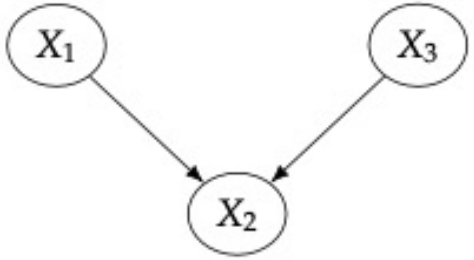



**(a)** Chain directed to the right

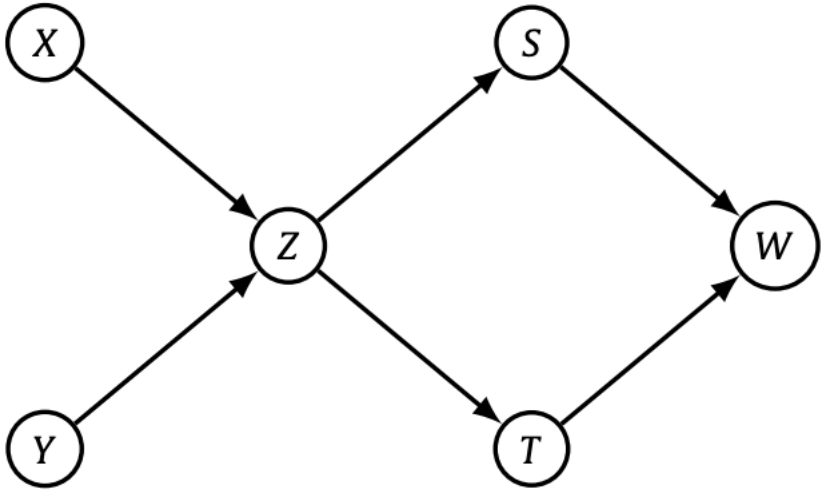
**(b)** Chain directed to the left

**(c)** Fork

**Figure 11.2:** Three Markov equivalent graphs



# An Example

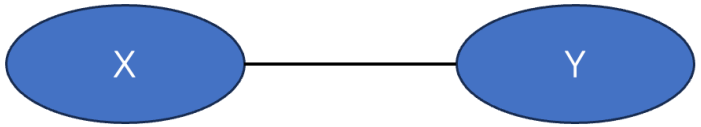



**In this figure, X and Y are d-separated without conditioning on Z , since they form a collider. The same does not hold for X and S, given that they form a chain by means of Z, and therefore conditioning (i.e. setting its value) on the middle vertex Z d-separates X from S.**

## Partially DAG.

The graph G is a partially-directed acyclic graph (PDAG) if it can contain both undirected (−) and directed (→) edges.

## Skeleton:



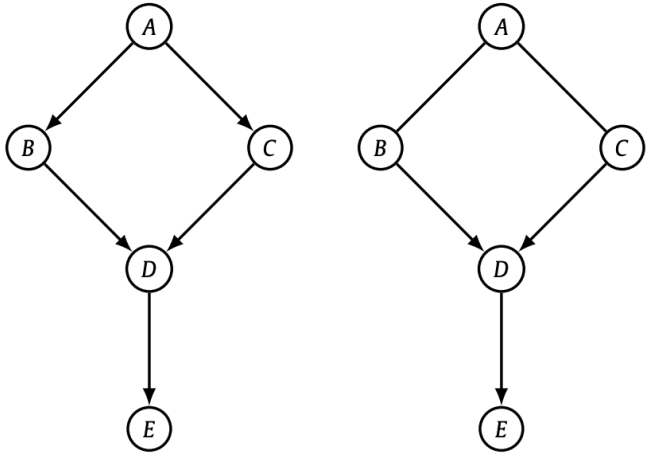

## V-structure:

Let G be a PDAG. A v-structure in G is a triple X → Y ← Z where X and Z are not adjacent. V-structures are also called unshielded colliders

## Observational equivalence:

Two DAGs G and H are observationally Markov equivalent if they have the same skeleton and the same v-structures, denoted as G ≡ H

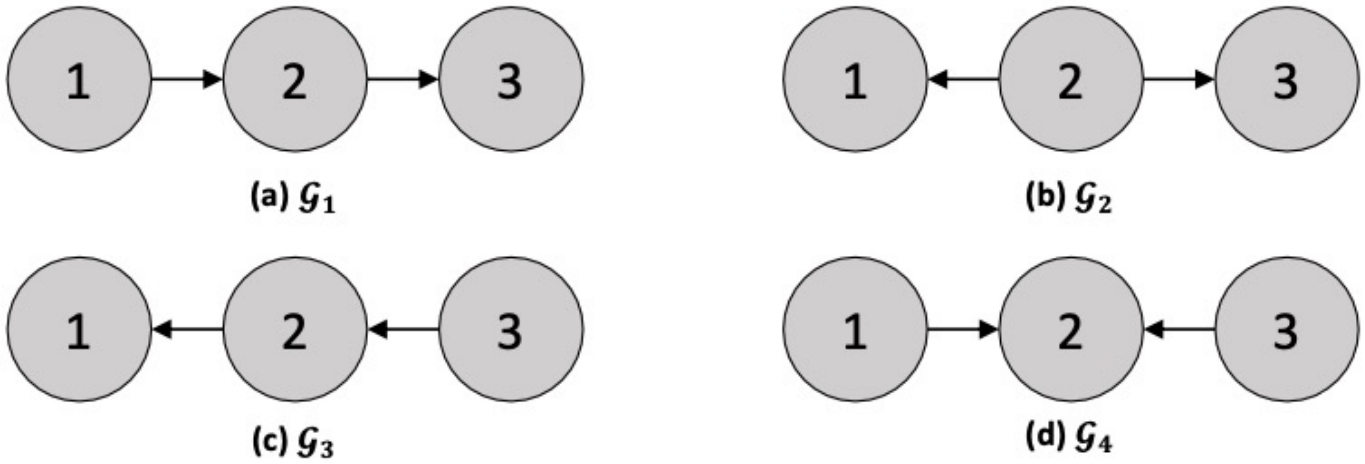
**Completed PDAG:** A PDAG G is said to be completed if any directed edge is compelled and any undirected edge is reversible w.r.t. its MEC [G]




**A DAG on the left and its CPDAG on the right.**
**As we can see, both graphs have the same underlying structure (i.e. skeleton), but differ from the orientation of some of the edges**.

Specifically, the
edges connecting A to B and C can be rearranged to form different chains or a fork. This is not true for the others edges in the CPDAG, since they are compelled. In fact, modifying the orientation of one of them would either remove the v-structure formed by B → D ← C or introduce a new one.

# Markov equivalence




(a), (b), (c) Three Markov equivalent graphs

## Markov Assumption

Markov assumption tells us if variables are d-separated in the graph $G$, then they are independent in the distribution $P$

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

However, going from independencies in the distribution $P$ to d-separations in the graph $G$ isn't something that the Markov assumption gives us, what we need is converse of Markov Assumption

**Assumption 11.1** (Faithfulness)

$$X \perp\!\!\!\perp_G Y \mid Z \impliedby X \perp\!\!\!\perp_P Y \mid Z \tag{11.1}$$

In addition to faithfulness, many methods also assume that there are no unobserved confounders, which is known as *causal sufficiency*.

**Assumption 11.2** (Causal Sufficiency) *There are no unobserved confounders of any of the variables in the graph.*

# Mixed graph

**The graph G is a mixed graph (MG) if it contains undirected (−), directed (→) and bidirected (<->) edges**

## M-separation

Let G be a MG, π be a path on G and Z a subset of V. The path π is blocked by Z if and only if π contains:

- a non-collider such that the middle vertex is in Z, or
- a collider such that middle vertex, or any descendant of it, is not in Z

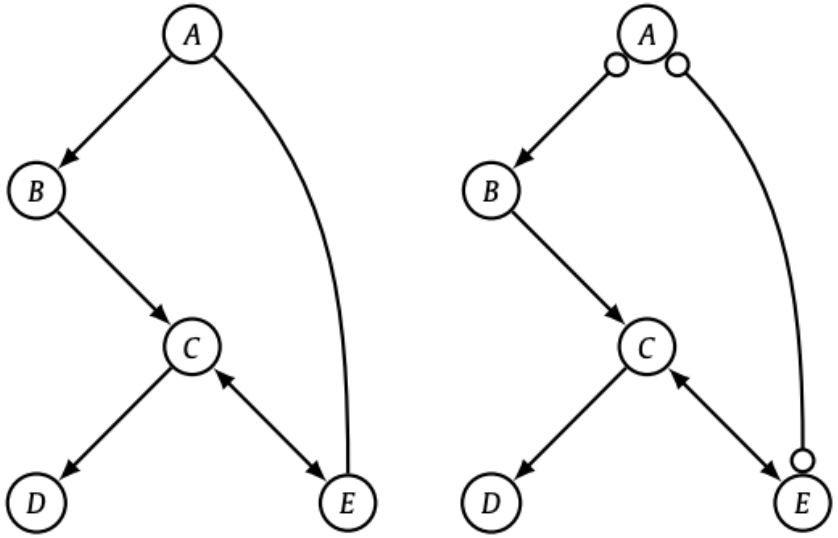
## Ancestral graph: A mixed graph G is ancestral if:

- $G$ has no (directed) cycles, and
- $X \in Sp(Y)$, then $X \notin An(Y)$, and
- $X \in Ne(Y)$, then $Pa(X) = \varnothing \wedge Sp(X) = \varnothing$.

**Maximal ancestral graph**. An ancestral graph is maximal (MAG) if any pair of non adjacent vertices are graphically separated (in terms of m-separation)

**Partial ancestral graph.** The graph G is a partial ancestral graph (PAG) if it contains any combination of the following edge marks: tail (−), arrowhead (→) and circle (◦). Moreover, let [G] be the MEC associated to G, then:

- $G$ has the same adjacencies of $[G]$, and
- any arrowhead mark in $G$ is invariant in $[G]$, and
- any tail mark in $G$ is invariant in $[G]$.




*X◦−◦Y:* Exactly one of the following holds:
X causes Y or vice versa; there is an unobserved confounder that causes X and Y ; or both (1) and (3) hold; or both (2) and (3) hold

A mixed graph on the left and one of its possible PAGs on the right

# Methods for Causal Structure Learning

# Methods

## Constraint-based methods

**Constraint-based methods are natural when viewing causal structure learning as a constraint satisfaction problem, where conditional independences or other constraints that can be inferred from data are used to iteratively prune the space of possible graphs.**

## Score-based methods

**In contrast, score-based methods arise from viewing causal structure learning as a combinatorial optimization problem.**

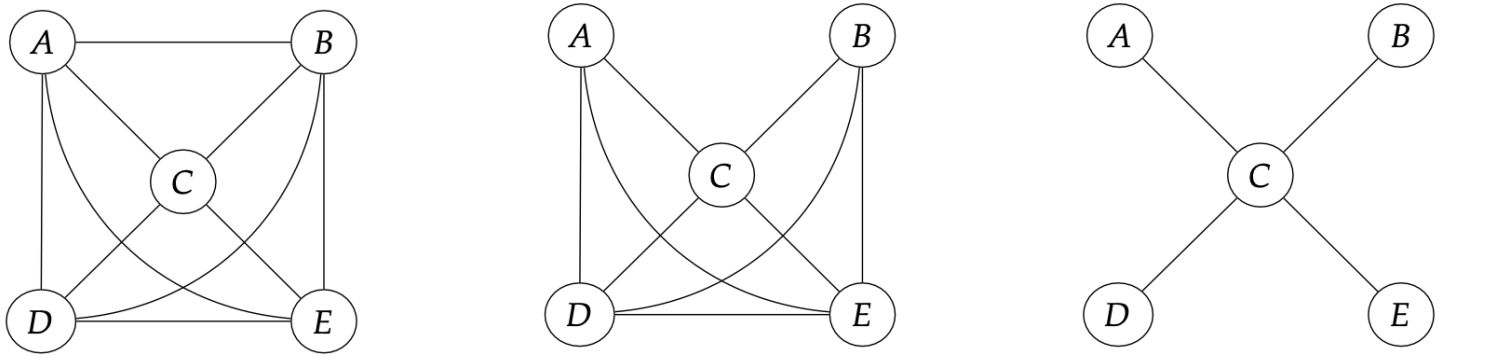
# Constrained based method

## The PC Algorithm

**PC starts with a complete undirected graph and then trims it down and orients edges via three step**

1. Identify the skeleton.
2. Identify immoralities and orient them.
3. Orient qualifying edges that are incident on colliders.

**1. Identify the skeleton**




**(a)** Complete undirected graph that we start with

**(b)** Undirected graph that remains after removing $X - Y$ edges where $X \perp\!\!\!\perp Y$

**(c)** Undirected graph that remains after removing $X - Y$ edges where $X \perp\!\!\!\perp Y \mid Z$

**Figure 11.7:** Illustration of the process of step 1 of PC, where we start with the complete graph (left) and remove edges until we've identified the skeleton of the graph (right), given that the true graph is the one in Figure 11.6.

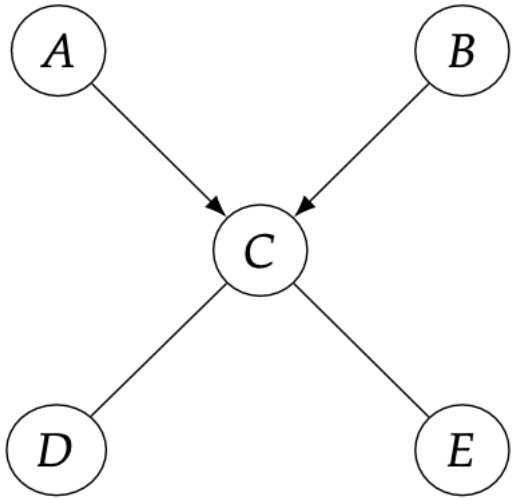



**Figure 11.8:** Graph from PC after we've oriented the immoralities.

## 2. Identifying the Colliders

[3] This is called *orientation propagation*.

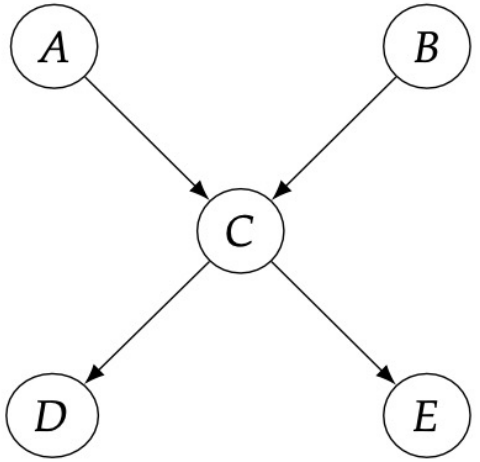



**Figure 11.9:** Graph from PC after we've oriented edges that would form immoralities if they were oriented in the other (incorrect) direction.

## 3. Orienting Qualifying Edges Incident on Colliders

# Score-based approaches

Score-based methods originated in parametric settings like discrete or linear Gaussian models.

**Score-based algorithms are usually structured around the maximization of a measure of fitness of a graph G through a space of possible graphs G for the observed samples D, following a defined scoring criterion S(G, D)**

$$G^* = \underset{G \in \mathbb{G}}{\arg\max}\, \mathscr{S}(G, \mathbf{D})$$

# Some important definitions

**Definition 3.4** *(Decomposable score).* A scoring criterion $\mathcal{S}(G, \mathbf{D})$ is *decomposable* if it can be defined as a sum of the scores over a vertex and its parents:

$$\mathcal{S}(G, \mathbf{D}) = \sum_{X \in \mathbf{V}} \mathcal{S}(X, Pa(X), \mathbf{D}) \tag{3.4}$$

**Definition 3.5** *(Equivalent score).* A scoring criterion $\mathcal{S}(G, \mathbf{D})$ is *score equivalent* if $\mathcal{S}(G, \mathbf{D}) = \mathcal{S}(H, \mathbf{D})$, for each pair of graphs $G$ and $H$ in the same equivalence class.

**Definition 3.6** *(Consistent score).* Let $\mathbf{D}$ be a dataset associated with a probability distribution $P$, and let $G$ and $H$ be two graphs. A scoring criterion $\mathcal{S}$ is said to be *consistent* in the limit of the number of samples if and only if:

- If only $G$ contains $P$, then $\mathcal{S}(G, \mathbf{D}) > \mathcal{S}(H, \mathbf{D})$,
- If both $G$ and $H$ contain $P$ and the model associated with $H$ has fewer parameters that the one with $G$, then $\mathcal{S}(G, \mathbf{D}) < \mathcal{S}(H, \mathbf{D})$.

# Score criterion

This property guarantees that any deletion of an unnecessary edge will produce a higher score value, allowing the definition of an optimal greedy search algorithm.

One of the most commonly used score criterion is the Akaike Information Criterion (AIC) [68]:

$$AIC = 2k - 2\ln\hat{L} \tag{3.5}$$

where $k$ is the number of parameters of the model and $\hat{L}$ is the maximum value of the likelihood for the given model. Models achieving a lower value of AIC are preferred, i.e. they explain better the observed data. Another common scoring criterion is offered by the Bayesian Information Criterion (BIC) [69], also known as the Schwarz Information Criterion:

$$BIC = k\ln n - 2\ln\hat{L} \tag{3.6}$$

which differs from AIC due to the parameters penalty term that takes into account the number of observations $n$. Others commonly used scoring criteria are the Bayesian Dirichlet equivalent uniform (BDeu) [70] and the Bayesian Dirichlet sparse (BDs) [71].

A: 1, 2, 3, 4, 5
B: 5, 4, 3, 2, 1

**Based on the AIC scores, the model A → B has the smallest AIC and thus is the preferred model. This would suggest that A is causing B.**

**1. A → B Model**

Consider a linear model $B = \alpha + \beta A$. Using least squares, we can determine the parameters $\alpha$ and $\beta$. For this data, the best fit line is $B = 6 - A$ giving $\alpha = 6$ and $\beta = -1$.

We can compute the likelihood $L$ and subsequently the AIC. For simplification, let's assume the AIC value is 10 (actual likelihood computation would involve assumptions of normality and further math).

**2. B → A Model**

Similarly, consider a linear model $A = \gamma + \delta B$. Determining the parameters for this direction might be trickier since it's not the true relationship of the data. For simplification, let's assume its AIC value is 20.

**3. A - B Model (Independent)**

We could assume that A and B are independent, with no relationship. The AIC value for this model might be very large since it won't explain the data well. For simplification, let's assume its AIC value is 30.

# Learning DAGs using permutation-based algorithms

## Greedy Sparsest Permutation(GSP)

**A hybrid method that constrains the search space to the set of (estimated) minimal I-MAPs of PX.**

**Theorem 1 (from [173])** *Given a permutation $\pi$ and a distribution $\mathbb{P}_X$, there exists a unique graph $\mathcal{G}_{\mathbb{P}_X}(\pi)$ that is consistent with $\pi$ and is a minimal I-MAP for $\mathbb{P}_X$. This graph has edges*

$$\{i \to j \mid X_i \not\perp\!\!\!\perp X_j \mid X_{\mathrm{pre}_\pi(j)\setminus\{i\}}\} \qquad \textit{where } \mathrm{pre}_\pi(j) = \{k \mid k <_\pi j\}.$$

Since the minimal I-MAPs of $P_X$ are the (locally) sparsest DAGs which can correctly model $P_X$, they form a natural space over which to search for the true DAG $G_*$.

For instance, consider a permutation of four variables [A, B, C, D]. This means that in a DAG consistent with this permutation, B doesn't direct towards A, C doesn't direct towards A or B, and D doesn't direct towards A, B, or C.

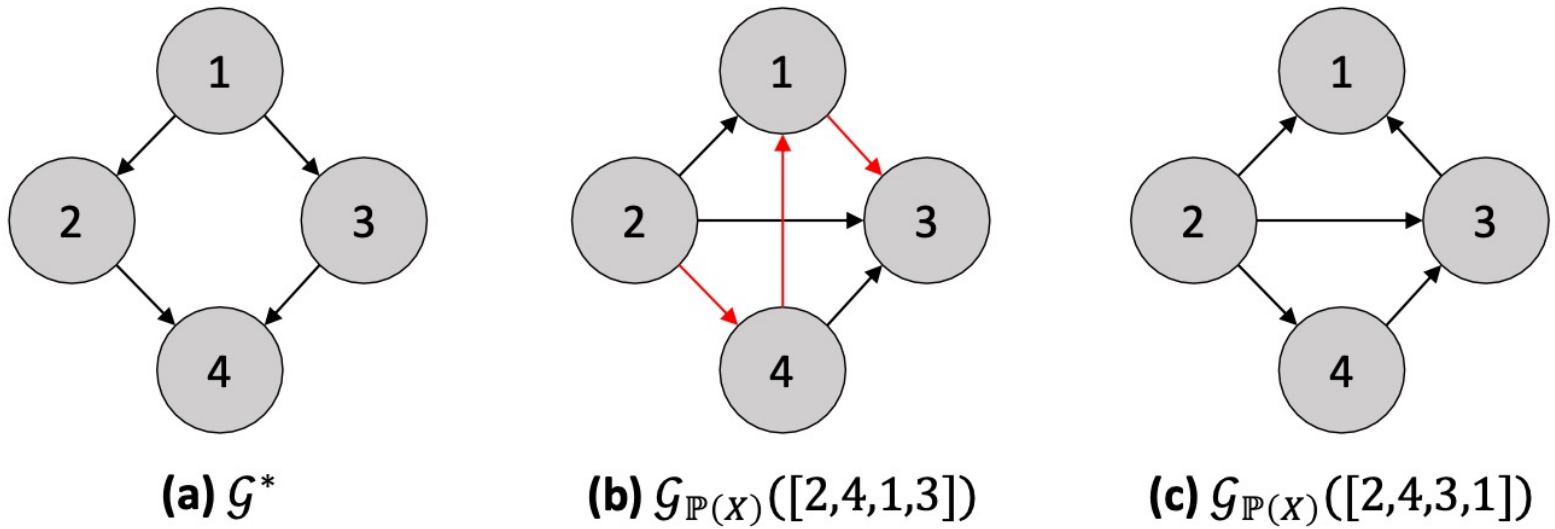



Figure 7: **A greedy step over minimal I-MAPs performed by GSP.** (a) The true graph $\mathcal{G}^*$, to which the distribution $\mathbb{P}(X)$ is faithful. (b) The minimal I-MAP associated with the permutation $\pi^{(0)} = [2, 4, 1, 3]$, with covered edges shown in red. (c) The minimal I-MAP associated with the permutation $\pi^{(1)} = [2, 4, 3, 1]$, obtained after flipping the covered edge $1 \to 3$.

# FCM-Based Approaches

**The two families of methods above either face the inseparability of the MEC or the need for large samples to confirm causal faithfulness.**

**Causal discovery can also be conducted based on Functional Causal Models (FCM), which is also known as SCM and describes a causal system via a set of equations.**

In FCM, each variable is explained by an equation in terms of its direct causes and some additional noise. For example, the function $x_j = f_j(x_i, u_j)$ explains the causal link $x_i \rightarrow x_j$ with some additional noise $u_j$.

**LiNGAM is a typical FCM-based causal discovery algorithm in non-temporal setting**

# Linear Non-Gaussian Acyclic Model (LiNGAM)

In the context of linear causal models, when causal sufficiency (Definition 2.24) holds, the observed variables can be expressed as a linear combination of the noise terms:

$$\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{e} \tag{3.8}$$

$$\mathbf{x} = (\mathbf{I} - \mathbf{B})^{-1}\mathbf{e} = \mathbf{A}\mathbf{e}$$

Assumption:

Causal sufficiency

There exist a linear generation process between x and y.

The noise e_i are independent from each other.

The noise e_i will follow non gaussian distribution.

**Models that meet the above criteria can be solved using an algorithm for LiNGAM, and an ICA-based algorithm is described next.**

# Independent component analysis(ICA)

**ICA, also known as blind signal processing, was originally used to separate a source signal from multiple mixed signals.**

**Similar to PCA, the purpose of ICA is to convert the signals of an incoming system into signals that are independent of each other.**

**Unlike PCA, which aims to separate the signals with the highest variance for data compression, ICA is based on the non-Gaussian assumption and separates the source signals that have the same variance and follow a non-Gaussian distribution.**

**As mentioned before, LiNGAM is based on a linear data. We have linear model:**

$$x_i = \sum_{k(j)<k(i)} b_{ij} x_j + e_i + c_i$$

**Transform from the equation**

$$\boldsymbol{x = Bx + e}$$

**B is the matrix representing causal weight, solving for x, we have:**

$$\mathbf{x = (I - B)^{-1} e = Ae}$$

**According to the assumption of non_gaussian, we have the only solution of A and its inverse matrix $W = A^{-1}$ using ICA**

$$\boldsymbol{W = I - B}$$

# Causal discovery In Time Series

# 1. Causal discovery

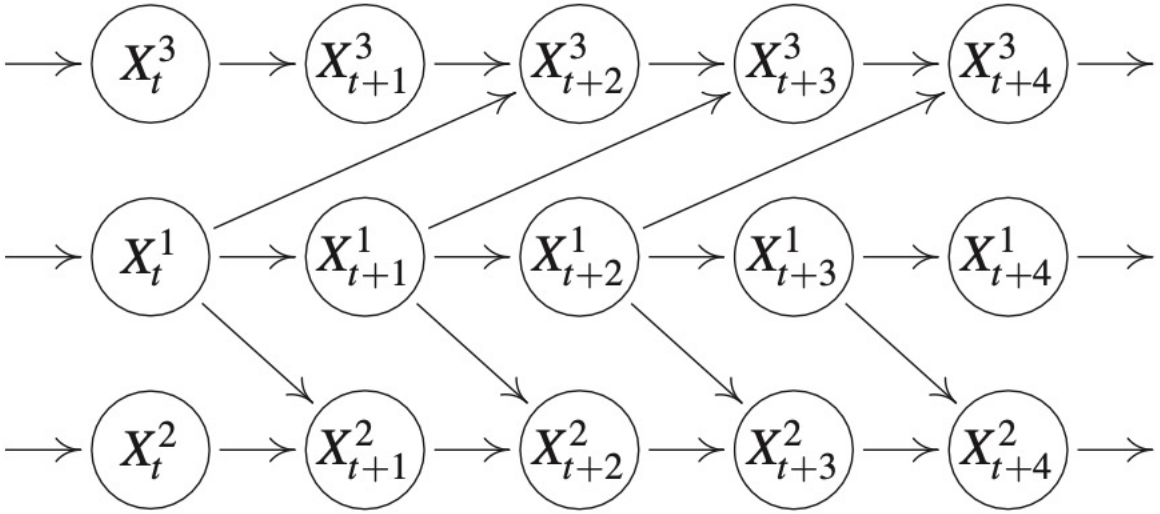



Figure 10.1: Example of a time series with no instantaneous effects.

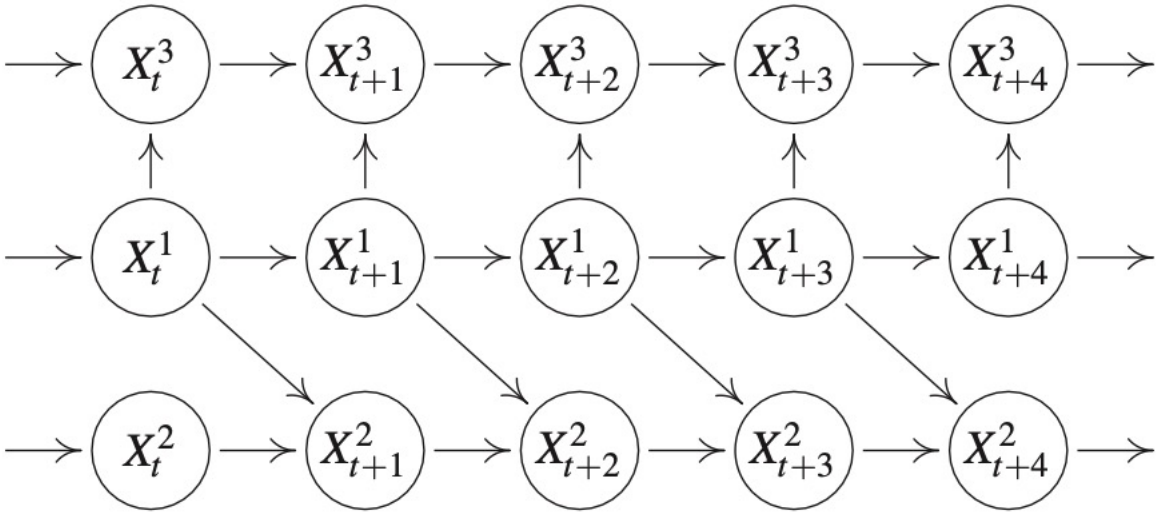
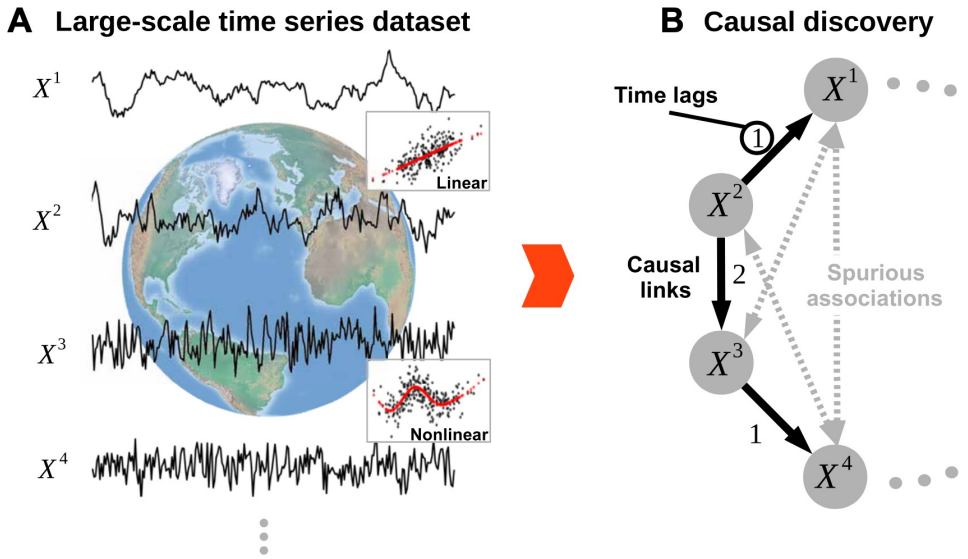



Figure 10.2: Example of a time series with instantaneous effects.

# PCMCI



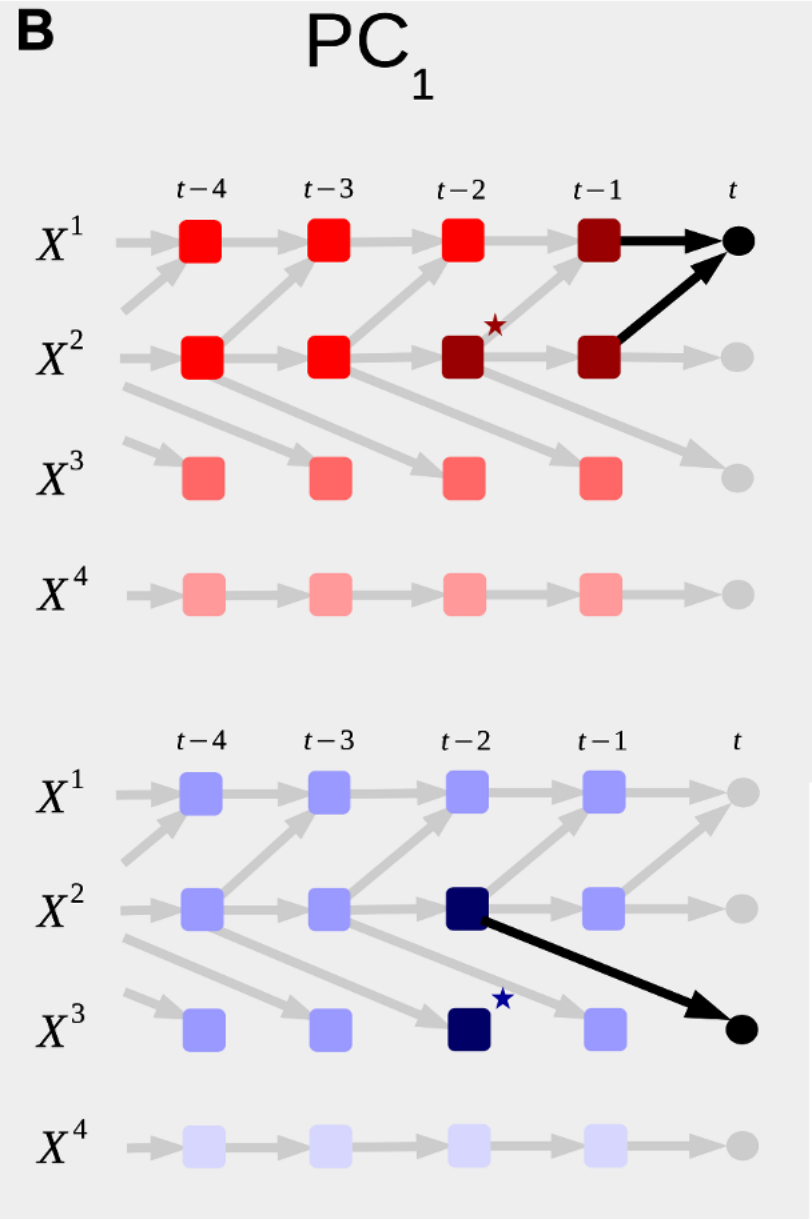

**Consider an underlying time-dependent system**

$$X_t^j = f_j(\mathcal{P}(X_t^j), \eta_t^j)$$

**(1) PC1 condition selection to identify relevant conditions for all included time series variables**

**(2) the momentary conditional independence (MCI) test to test whether $X_{t-\tau}^i \rightarrow X_t^j$ with**

$$\text{MCI:} \quad X_{t-\tau}^i \perp\!\!\!\perp X_t^j \mid \widehat{\mathcal{P}}(X_t^j) \setminus \{X_{t-\tau}^i\}, \widehat{\mathcal{P}}(X_{t-\tau}^i).$$




**Let $V$ be the set of all variables, and $X$ be the target variable.**

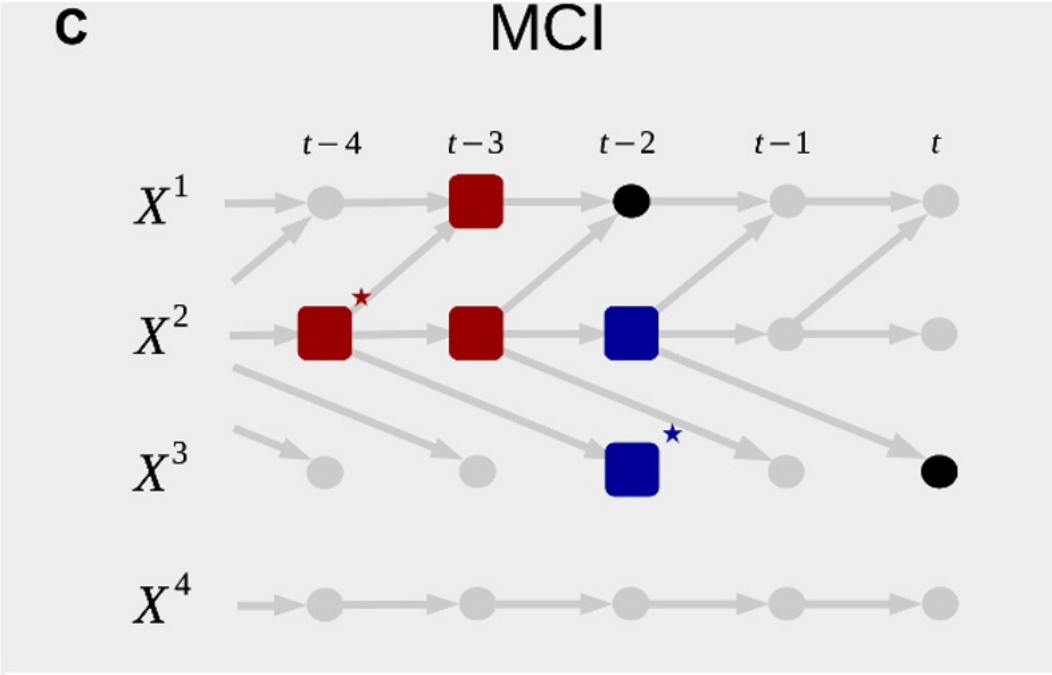
**1.Initialization:**

$V'$= {variables in $V$ that are not unconditionally independent of $X$}

**2.For each iteration *i*:**

a. Identify $i$ = variable in $V'$ with the strongest dependency on $X$ from the prior iteration.

b. Update $V'$= $V'$ - {variables in $V'$ that are independent of $X$ given $Yi$}



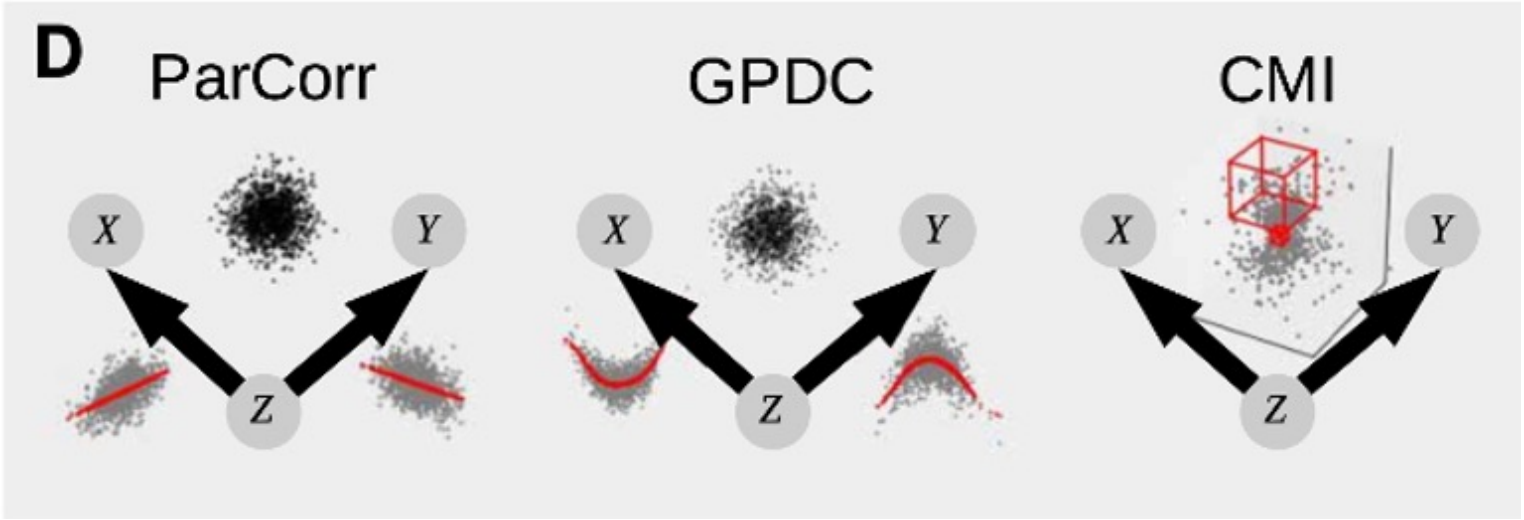

# Momentary Conditional Independence(MCI)

1. Conditional independence
2. Considering Time Step
3. Autocorrelation

$$\mathrm{MCI}: X^i_{t-\tau} \not\perp\!\!\!\perp X^j_t \mid \widehat{\mathcal{P}}\left(X^j_t\right) \setminus \left\{X^i_{t-\tau}\right\}, \widehat{\mathcal{P}}\left(X^i_{t-\tau}\right)$$

**These low-dimensional conditions are then used in the MCI conditional independence test:**

For testing $X^1_{t-2} \rightarrow X^3_t$ , the conditions $\hat{P}(X^3_t)$ (blue boxes) are sufficient to establish conditional independence, while the additional conditions on the parents $\hat{P}(X^1_{t-2})$ (red boxes) account for autocorrelation and make MCI an estimator of causal strength.




•The gray scatter plots depict regressions of X and Y based on Z.
•The black scatter plots represent the residuals of these regressions.
•The red cubes in the context of CMI symbolize the k-nearest neighbor test, which operates adaptively with the data without requiring an additivity assumption.

**Linear vs. Nonlinear**:
•**ParCorr (Partial Correlation)**: A linear independence test that *assumes linear additive noise models*.
•**GPDC (Generalized Partial Directed Coherence)**: A nonlinear test, but it makes an *assumption of additivity, not delving deeper into other nonlinear relationships*.
•**CMI (Conditional Mutual Information)**: Another nonlinear test. This one employs a data-adaptive*, model-free k-nearest neighbor technique*.

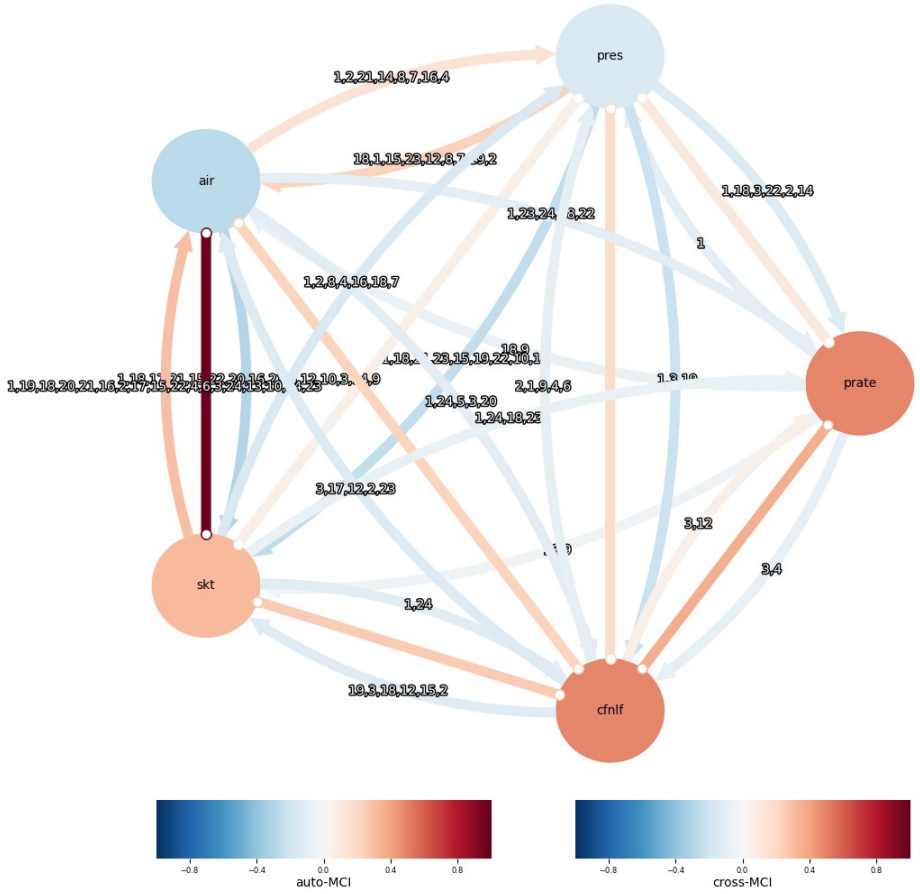
# PCMCI+

## Separate Skeleton Edge Removal into Two Phases:

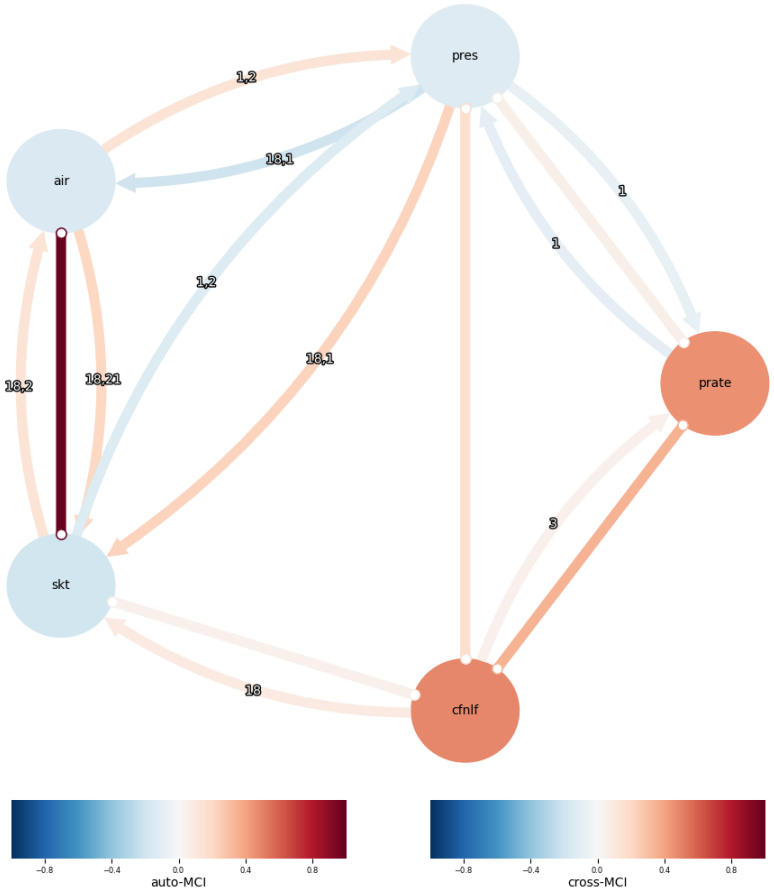
## 1. Lagged Conditioning Phase
## 2. Contemporaneous Conditioning Phase

**Adaptive Testing Strategy:** Instead of applying a uniform test to all possible links, PCMCI+ adaptively selects the most appropriate CI test for each link based on the characteristics of the data. This is crucial because, in complex datasets, not all relationships are of the same type; some might be linear, while others could be nonlinear.

# PCMCI


pres
air
prate
skt
cfnlf
1,2,21,14,8,7,16,4
1,18,3,22,2,14
1,2,3,4,16,18,7
1,24,5,3,20
1,24,18,2
2,1,9,4,6
3,17,12,2,23
3,12
3,4
1,24
19,3,18,12,15,2
auto-MCI
cross-MCI

# PCMCI+


pres
air
prate
skt
cfnlf
1,2
18,1
1
1,2
18,2
18,21
18,1
3
18
auto-MCI
cross-MCI

# Problem

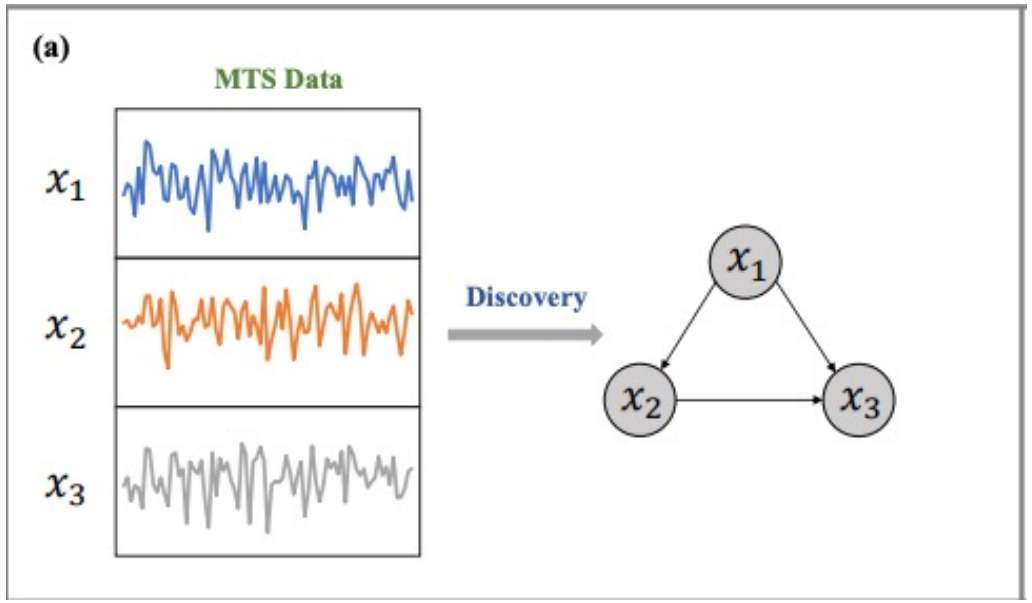
The time series forecasting problem can be formulated as:

***Predicting future M steps(day or month) precipitation*** $Y_{t+1:t+M}$***. Given the previous L steps of observations*** $Y_{t-L:t}$ ***and the covariates(air temperature … )*** $X_{t-L+1:t}$ **.**

***Input:*** $Y_{t-L:t}$***,*** $X_{t-L+1:t}$

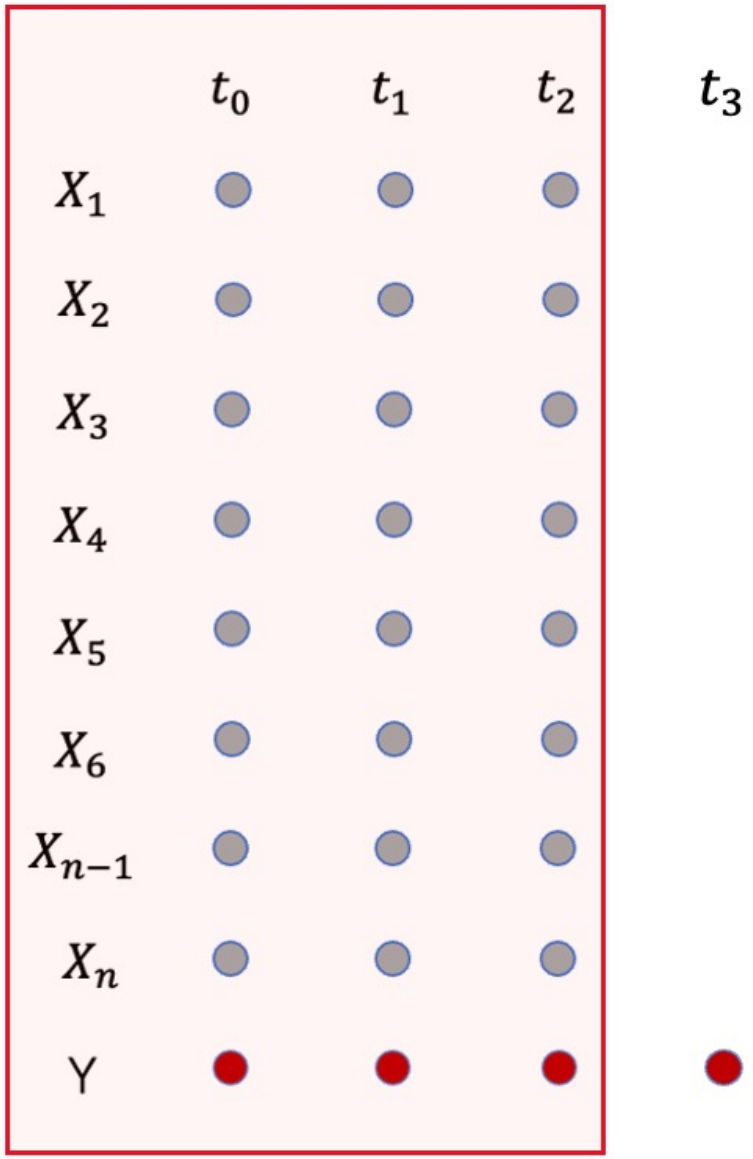
***Output:*** $Y_{t+1:t+M}$

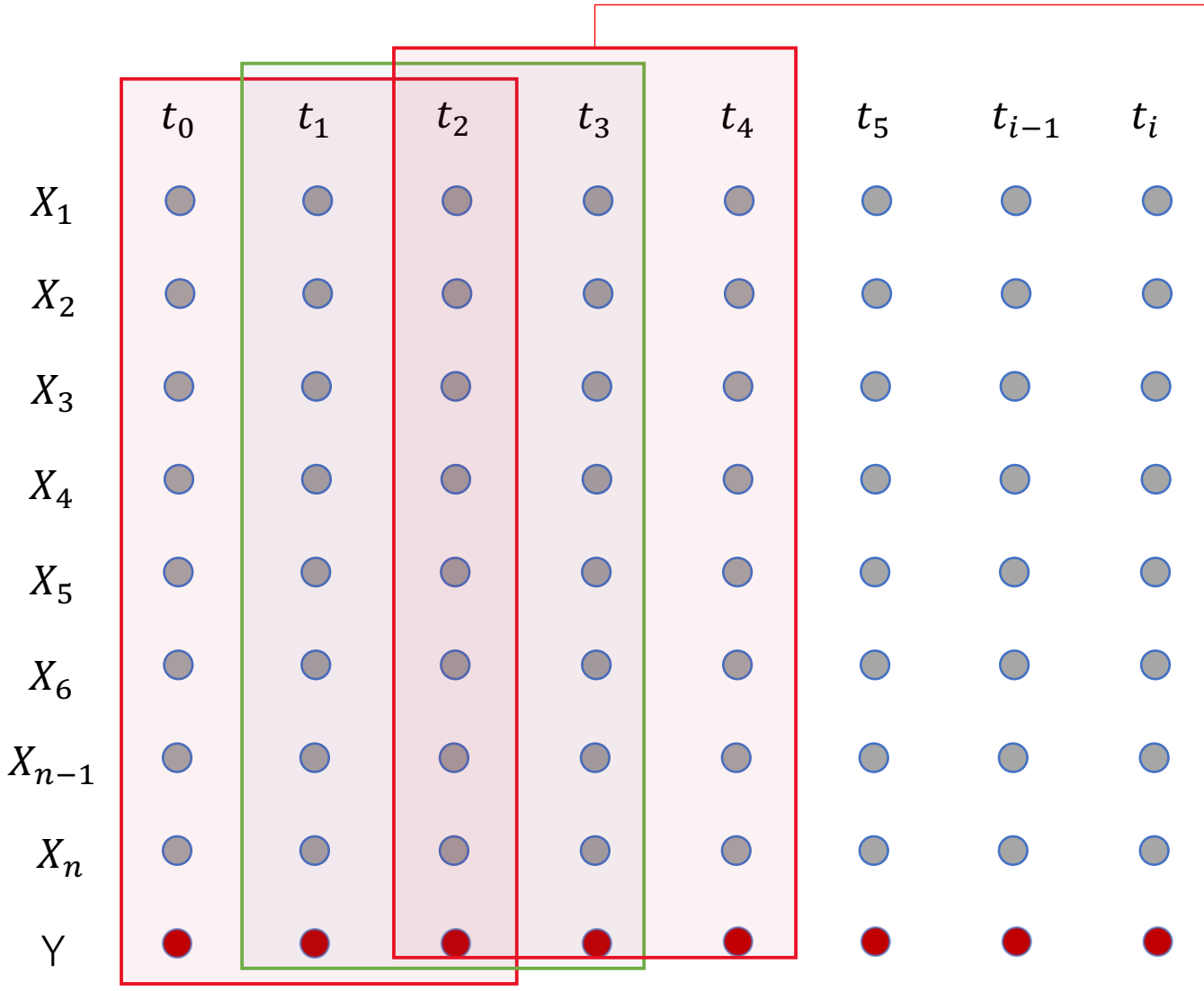
# Defining causal problem in time series

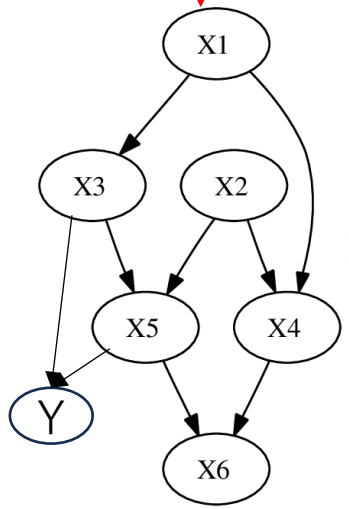



Our data: Linear and Nonlinear relations

Hidden confounders






✖ N(Number of windows)

For every time step window, we learnt a causal structure graph.

# Next Month

- **Reading more about ICA based causal structure learning.**
- **Writing Research proposal(Intro and Literature review)**
- **Thinking about how to use latent variables to help prediction**