BUILD with Precision: Bottom-Up Inference of Linear DAGs

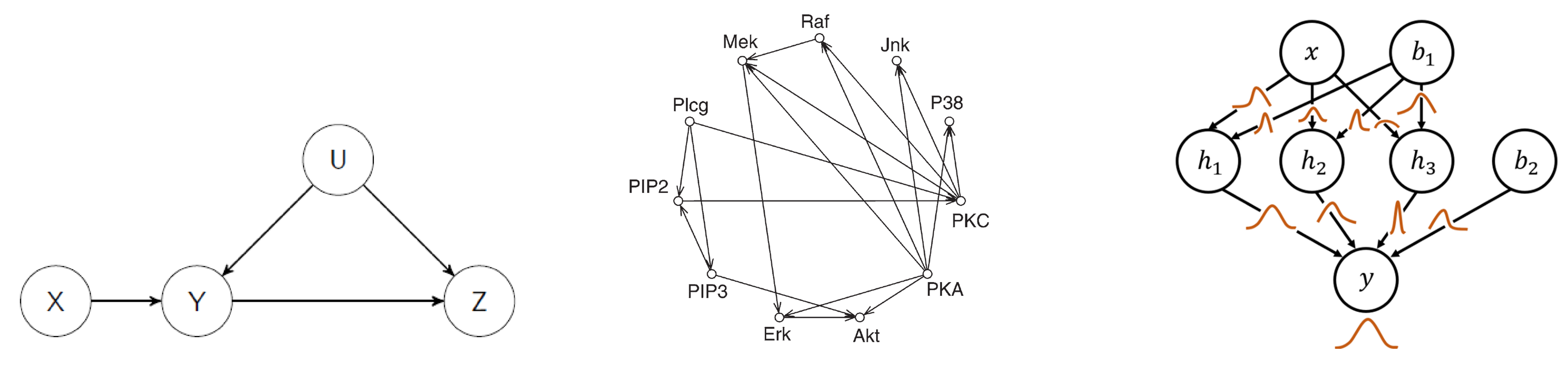
Hamed Ajorlou[1], **Samuel Rey**[2], Gonzalo Mateos[1], Geert Leus[3], Antonio G. Marques[2]

[1]University of Rochester [2]King Juan Carlos University [3]TU Delft

hajorlou@ur.rochester.edu

Motivation, context, and goal

- **Directed acyclic graphs (DAGs)** have become prominent models in ML applications
  - ⇒ DAG edges may have causal interpretations [Peters17]
  - ⇒ Conditional independencies exist among variables in Bayesian networks
- DAGs appear in a gamut of applications: biology, genetics, and finance [Sachs05]
  - ⇒ The structure of the DAG is often unknown or unavailable



Causal inference | Protein signaling | Bayesian networks

- Learning DAGs from observation is challenging due to acyclicity
  - ⇒ Combinatorial search is NP-hard, scales super-exponentially
  - ⇒ Continuous optimization (NOTEARS, DAGMA): non-convex landscapes [Zheng18]
  - ⇒ Order-based (TOPO, Daskalakis et al.): may neglect edge weights

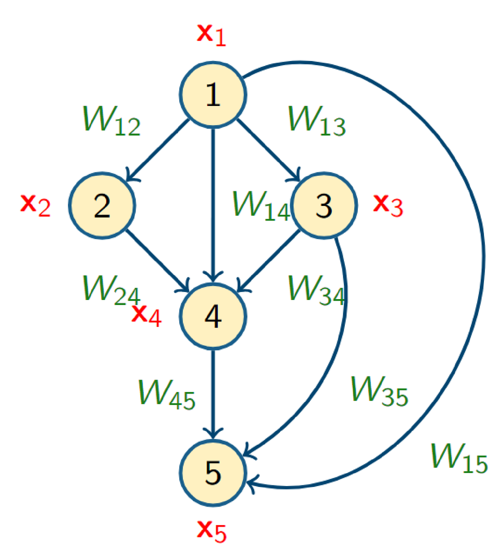
Our contribution: a **deterministic**, **bottom-up** algorithm that leverages the **precision matrix** to recover DAGs **without** optimization

Preliminaries: DAGs and linear SEM

- A DAG $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ is a set $\mathcal{V}$ of N nodes and a set of edges $\mathcal{E}$
  - ⇒ The adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ encodes its connectivity
  - ⇒ The entry $A_{ij} \neq 0$ indicates a directed link $j \to i$
- Define a graph signal $\mathbf{x} \in \mathbb{R}^N$ whose properties depend on $\mathcal{D}$
  - ⇒ x_i depends on its parents $PA_i = \{j \in \mathcal{V} : A_{ij} \neq 0\}$
- Structural equation model (SEM) widely used in causal inference
  - ⇒ A linear SEM generates the signals $\mathbf{X} \in \mathbb{R}^{N\times M}$ according to

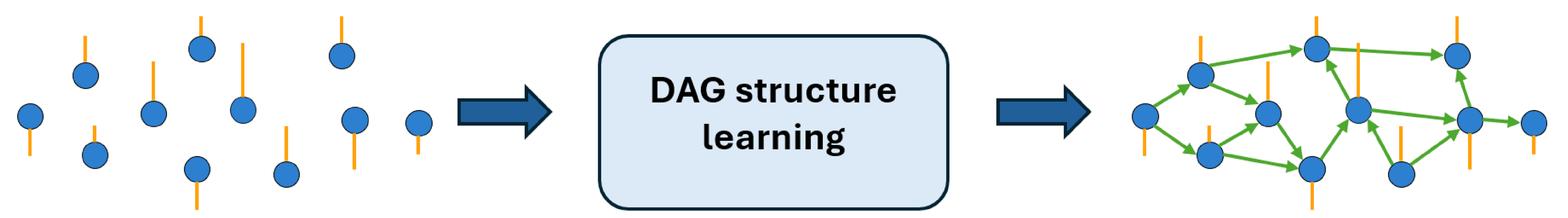
$$\mathbf{X} = \mathbf{A}\mathbf{X} + \mathbf{Z}$$

  - ⇒ Exogenous input $\mathbf{Z}$ is a random variable with diagonal covariance



DAG structure learning

- Given data $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_M] \in \mathbb{R}^{N\times M}$, adhering to a linear SEM determined by the DAG $\mathcal{D}$
  - ⇒ Learn the adjacency matrix $\mathbf{A}$ by solving a score-minimization problem



- **Key idea:** the unknown partial order of the nodes reflects on the covariance of the data
  - ⇒ Leverage this particular structure to recursively identify and prune leaves

Key Insight: Precision Matrix Encodes the DAG

- We focus on the linear Gaussian SEM setting: $\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{z}$, $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_N)$
  - ⇒ Covariance matrix: $\boldsymbol{\Sigma} = \sigma^2(\mathbf{I}_N - \mathbf{A})^{-1}(\mathbf{I}_N - \mathbf{A}^\top)^{-1}$
  - ⇒ Precision matrix: $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1} = \sigma^{-2}(\mathbf{I}_N - \mathbf{A} - \mathbf{A}^\top + \mathbf{A}^\top\mathbf{A})$

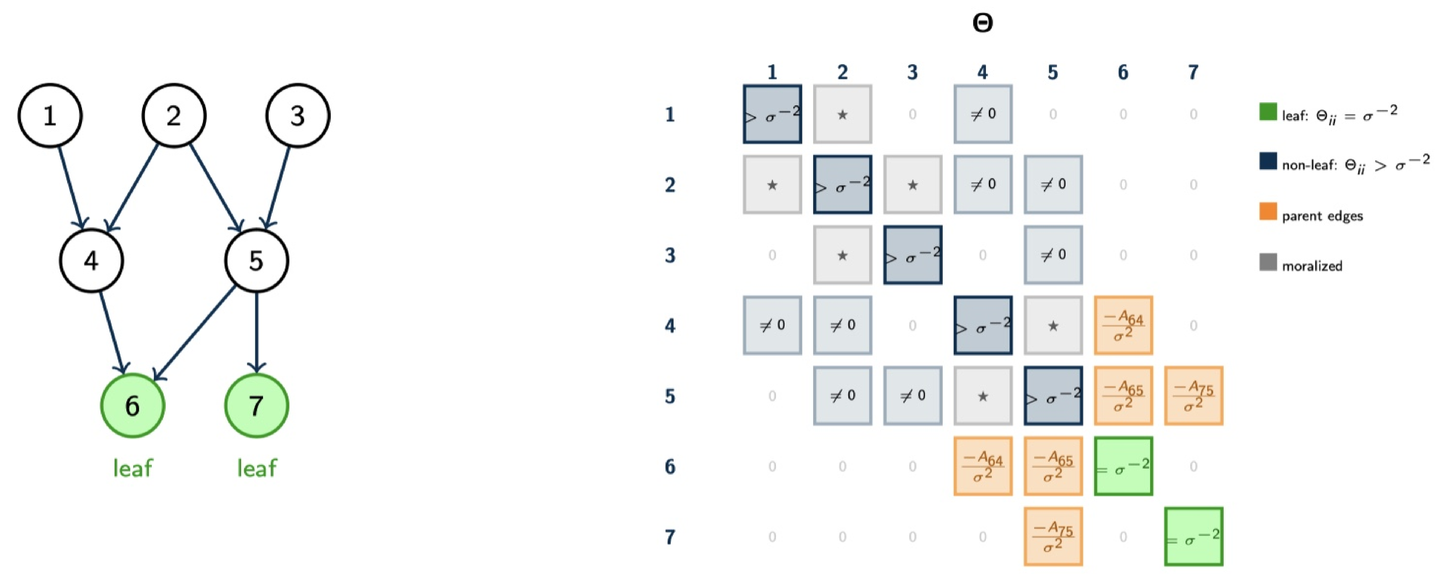
Lemma 1 (Precision matrix entries)

Let $\mathrm{CH}_j = \{i \in \mathcal{V} : A_{ij} \neq 0\}$ be the children of j. Then:

$$\sigma^2\Theta_{ij} = \begin{cases} 1 + \sum_{k\in\mathrm{CH}_i} A_{ki}^2, & i = j \\ -A_{ij} + \sum_{k\in\mathrm{CH}_i\cap\mathrm{CH}_j} A_{ki}A_{kj}, & i \neq j \end{cases}$$

- **Upshot:** Node $i \in \mathcal{V}$ is a leaf of $\mathcal{D}$ if and only if $\Theta_{ii} = \sigma^{-2}$
  - ⇒ All non-leaf nodes have $\Theta_{jj} > \sigma^{-2}$
  - ⇒ We can **identify leaves by inspection** of the precision matrix!

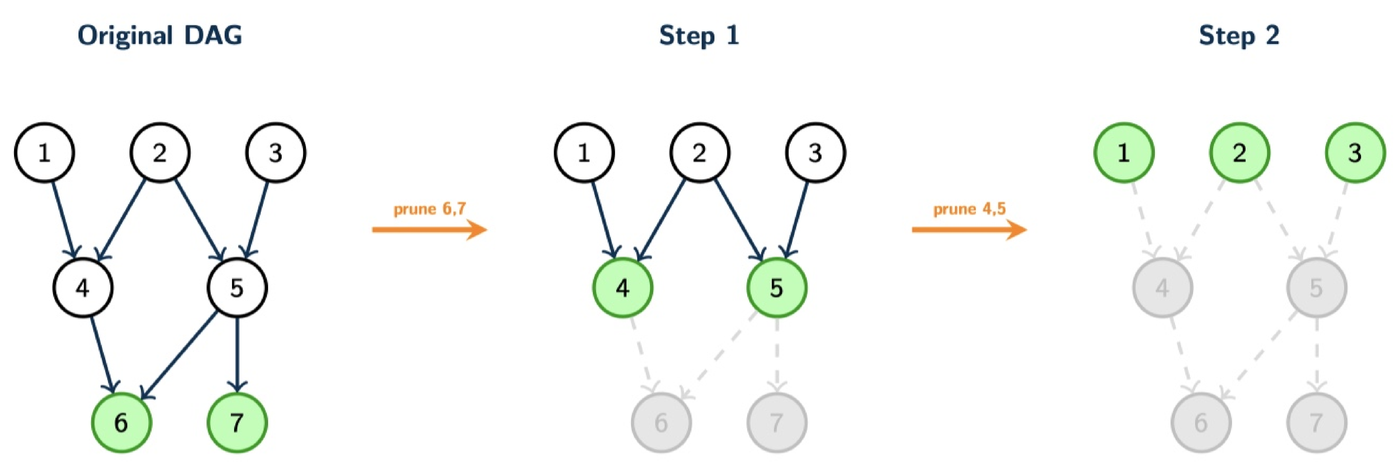
Example: Reading the DAG from Θ



- **Leaf rows** of **Θ reveal** parent identities and weights: $A_{ij} = -\sigma^2\Theta_{ij}$
  - ⇒ Moralized entries (⋆) arise from shared children, not direct edges

The BUILD Algorithm: Iterative Leaf Peeling

- **Idea:** given the ensemble precision **Θ**, we can identify and prune leaf nodes
  - ⇒ **Θ** is not known in practice and needs to be estimated



- The proposed BUILD algorithm comprises 2 phases:
  - ⇒ **Phase 1:** Estimate $\hat{\boldsymbol{\Theta}}$ once from data via GreedyPrune ($\mathcal{O}(N^3)$ cost)
  - ⇒ **Phase 2:** Iterative bottom-up DAG recovery
- The complexity of Phase 2 is $\mathcal{O}(N^2)$: find leaf $\mathcal{O}(N)$, recover parents $\mathcal{O}(N)$, deflate via Schur complement $\mathcal{O}(N^2)$

DAG recovery and refreshing

DAG recovery

Guarantee: Given the ensemble **Θ**, BUILD **exactly** recovers **A**

- The output is guaranteed to be a DAG by construction with no post-processing needed

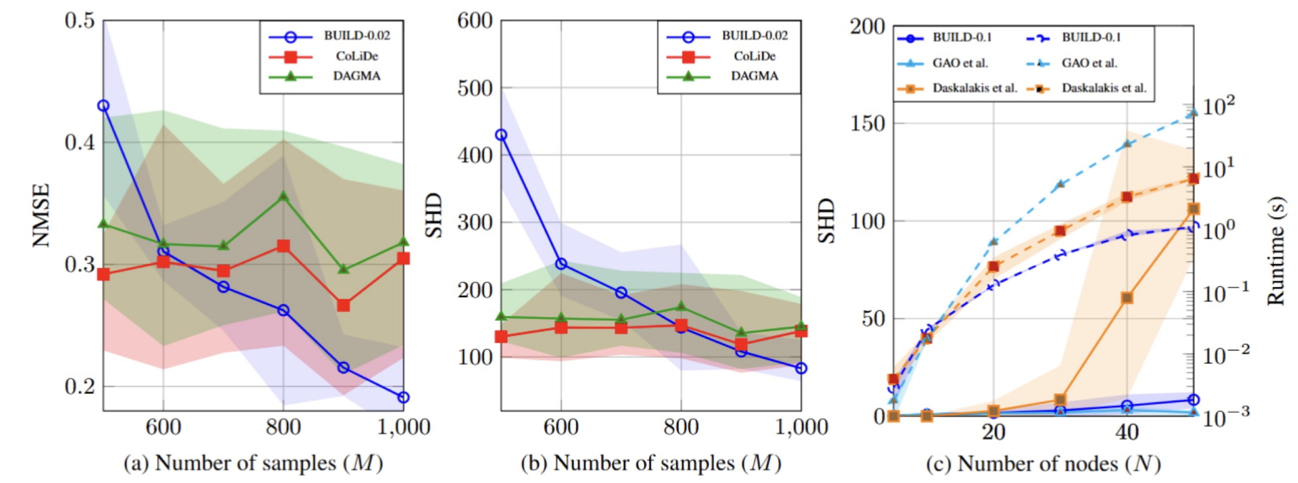
Refreshing

- Errors in estimating **Θ** may lead to errors in identifying leaves with cumulative effects
  - ⇒ BUILD-ρ re-estimates $\hat{\boldsymbol{\Theta}}$ on remaining nodes every $\rho \times 100\%$ of pruned leaves
  - ⇒ Provides an explicit **accuracy–runtime trade-off**

Results: ER-4 Graphs ($N = 200$, $M = 1{,}000$)

- Erdős–Rényi DAGs, weights $A_{ij} \sim U(-2, -0.5) \cup U(0.5, 2)$, Gaussian exogenous noise

Method	SHD ↓	FDR ↓	TPR ↑	Time (s) ↓
BUILD-0.005	**17.4 ± 3.6**	**0.004 ± 0.003**	**0.983 ± 0.004**	1203
BUILD-0.01	45.2 ± 10.9	0.035 ± 0.012	0.981 ± 0.004	621
BUILD-0.02	81.7 ± 28.5	0.072 ± 0.029	0.977 ± 0.004	324
BUILD-0.04	122.9 ± 34.0	0.112 ± 0.030	0.971 ± 0.007	168
CoLiDE	114.4 ± 43.1	0.031 ± 0.026	0.888 ± 0.031	109
DAGMA	136.0 ± 36.2	0.035 ± 0.021	0.864 ± 0.027	**94**



- BUILD outperforms CoLiDE and DAGMA once $M \geq 800$ samples are available

Results: Sachs Protein Signaling Network

- Cell-signaling data: $N = 11$ proteins, $M = 853$ samples, 17 experimentally known edges

	GOLEM-EV	GOLEM-NV	DAGMA	SortN Regress	DAGuer-reotype	GES	CoLiDE-EV	CoLiDE-NV	**BUILD**
SHD	22	15	16	13	14	13	13	12	**12**
FDR	0.83	0.66	0.50	0.61	0.57	0.50	0.54	0.53	**0.30**
TPR	0.11	0.11	0.05	0.29	0.17	0.23	0.29	0.35	**0.41**

- BUILD achieves **best SHD**, **lowest FDR**, and **highest TPR**

References

Peters17 J. Peters, D. Janzing, and B. Scholkopf. "Elements of Causal Inference: Foundations and Learning Algorithms". MIT Press, 2017.

Zheng18 X. Zheng, B. Aragam, P. K. Ravikumar, and E. P. Xing. "DAGs with NO TEARS: Continuous optimization for structure learning". Neurips, 2018.

Sachs05 K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan. "Causal protein-signaling networks derived from multiparameter singlecell data". Science, 2005.

Bello22 K. Bello, B. Aragam, and P. Ravikumar. "DAGMA: Learning DAGs via M-matrices and a log-determinant acyclicity characterization". NeurIPS, 2022.

Daskalakis25 C. Daskalakis, V. Kandiros, and R. Yao. "Learning Gaussian DAG models without condition number bounds". Int. Conf. Mach. Learning, 2025.