BUILD with Precision: Bottom-Up Inference of Linear DAGs

Gonzalo Mateos

Dept. of Electrical and Computer Engineering and Goergen Institute for Data Science and AI
University of Rochester
gmateosb@ece.rochester.edu

Collaborators: Hamed Ajorlou, Samuel Rey, Geert Leus, and Antonio G. Marques

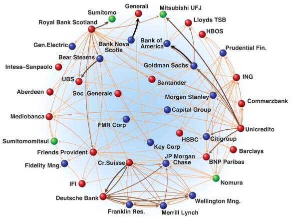
Acknowledgment: NSF Award ECCS-2231036, Spanish AEI PID2022-136887NB-I00

June 10, 2026

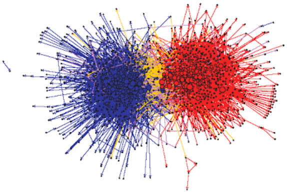
Learning from relational data

- Graphs are natural models for relational data that can help to learn in various timely applications

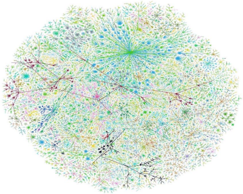
Economic Networks



Social and Information Networks

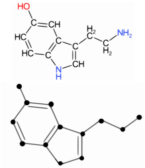


Internet

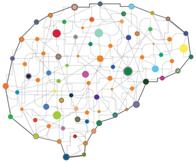


3D Meshes

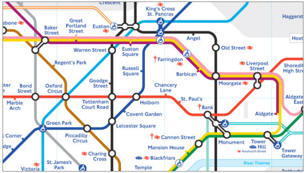
Molecules



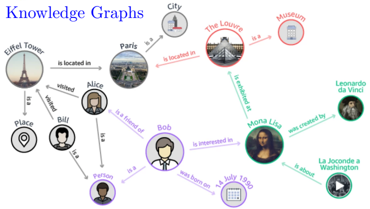
Brain Connectomes



Transportation Networks

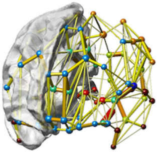


Knowledge Graphs



Learning graphs from data: Accounting for directionality and cycles

- Undirected topology inference from nodal observations [Kolaczyk'09]
  - Partial correlations and conditional dependence [Dempster'74]
  - Sparsity [Friedman et al'07] and consistency [Meinshausen-Buhlmann'06]



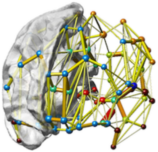
- Key in neuroscience and bioinformatics

  ⇒ Functional network from fMRI signals [Sporns'10]

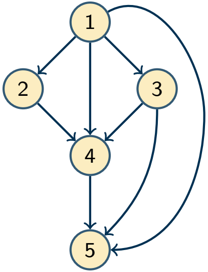
  ⇒ Gene-regulatory networks from microarray data [Mazumder-Hastie'12]

Learning graphs from data: Accounting for directionality and cycles

- Undirected topology inference from nodal observations [Kolaczyk'09]
  - Partial correlations and conditional dependence [Dempster'74]
  - Sparsity [Friedman et al'07] and consistency [Meinshausen-Buhlmann'06]
- Key in neuroscience and bioinformatics
  - ⇒ Functional network from fMRI signals [Sporns'10]
  - ⇒ Gene-regulatory networks from microarray data [Mazumder-Hastie'12]



- **This work:** learn the structure of directed acyclic graphs (DAGs)
- DAGs have become prominent models in various ML applications
  - ⇒ Conditional independences among variables in Bayesian networks
  - ⇒ DAG edges may have causal interpretations
  - ⇒ Bio [Sachs et al'05], genetics [Zhang et al'13], finance [Sanford-Moosa'12]
- **Challenges:** directionality, acyclicity (combinatorial constraint), identifiability



Roadmap

Background: Linear SEMs and DAG learning

Idea: Precision matrix structure and the BUILD algorithm

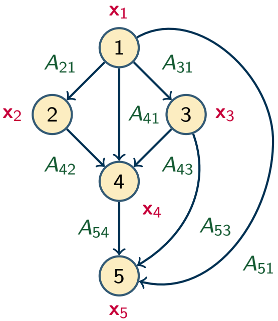
Experimental performance evaluation

Concluding remarks

H. Ajorlou *et al*, "BUILD with precision: Bottom-up inference of linear DAGs," *ICASSP*, 2026; see also arXiv:2512.16111 [cs.LG]

Linear structural equation (causal) models

- DAG $\mathcal{D}(\mathcal{V}, \mathcal{E}, \mathbf{A}) \in \mathbb{D}$, vertices $\mathcal{V} = \{1, \ldots, N\}$, edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$
  - $\Rightarrow$ Adjacency matrix $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_N] \in \mathbb{R}^{N \times N}$ of edge weights
  - $\Rightarrow$ Entry $A_{ij} \neq 0$ indicates a directed link from node j to i
- Random vector $\mathbf{x} = [x_1, \ldots, x_N] \in \mathbb{R}^N$, joint Markov $p(\mathbf{x})$ w.r.t. $\mathcal{D}$
  - $\Rightarrow$ DAG $\mathcal{D}$ encodes conditional independencies among variables in $\mathbf{x}$
  - $\Rightarrow$ Each x_i depends only on its parents $\text{PA}_i = \{j \in \mathcal{V} : A_{ij} \neq 0\}$

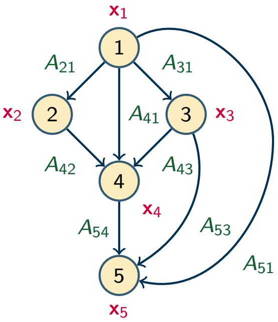


Linear structural equation (causal) models

- DAG $\mathcal{D}(\mathcal{V}, \mathcal{E}, \mathbf{A}) \in \mathbb{D}$, vertices $\mathcal{V} = \{1, \ldots, N\}$, edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$
  - $\Rightarrow$ Adjacency matrix $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_N] \in \mathbb{R}^{N \times N}$ of edge weights
  - $\Rightarrow$ Entry $A_{ij} \neq 0$ indicates a directed link from node j to i
- Random vector $\mathbf{x} = [x_1, \ldots, x_N] \in \mathbb{R}^N$, joint Markov $p(\mathbf{x})$ w.r.t. $\mathcal{D}$
  - $\Rightarrow$ DAG $\mathcal{D}$ encodes conditional independencies among variables in $\mathbf{x}$
  - $\Rightarrow$ Each x_i depends only on its parents $\text{PA}_i = \{j \in \mathcal{V} \, : \, A_{ij} \neq 0\}$
- Linear structural equation model (SEM) to generate $p(\mathbf{x})$

$$x_i = \sum_{j \in \text{PA}_i} A_{ij} x_j + z_i, \quad \forall i \in \mathcal{V}$$

  - $\Rightarrow$ Mutually independent, exogenous noises $\mathbf{z} = [z_1, \ldots, z_N]^\top$

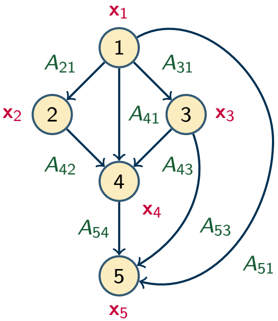


Linear structural equation (causal) models

- DAG $\mathcal{D}(\mathcal{V}, \mathcal{E}, \mathbf{A}) \in \mathbb{D}$, vertices $\mathcal{V} = \{1, \ldots, N\}$, edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$
  - $\Rightarrow$ Adjacency matrix $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_N] \in \mathbb{R}^{N \times N}$ of edge weights
  - $\Rightarrow$ Entry $A_{ij} \neq 0$ indicates a directed link from node j to i
- Random vector $\mathbf{x} = [x_1, \ldots, x_N] \in \mathbb{R}^N$, joint Markov $p(\mathbf{x})$ w.r.t. $\mathcal{D}$
  - $\Rightarrow$ DAG $\mathcal{D}$ encodes conditional independencies among variables in $\mathbf{x}$
  - $\Rightarrow$ Each x_i depends only on its parents $\text{PA}_i = \{j \in \mathcal{V} : A_{ij} \neq 0\}$
- Linear structural equation model (SEM) to generate $p(\mathbf{x})$

$$x_i = \sum_{j \in \text{PA}_i} A_{ij} x_j + z_i, \quad \forall i \in \mathcal{V}$$

  - $\Rightarrow$ Mutually independent, exogenous noises $\mathbf{z} = [z_1, \ldots, z_N]^\top$
- **Q:** Estimate $\mathbf{A}$ (learn DAG $\mathcal{D}$) using dataset $\mathbf{X} \in \mathbb{R}^{N \times M}$ with M i.i.d. samples from $p(\mathbf{x})$?



Innovation in context

- Learning a DAG solely from observational data **X** is NP-hard [Chickering'96]
- **Combinatorial search:** GES [Chickering'02], PC algorithm [Spirtes et al'00]
  - $\Rightarrow$ Scale poorly: $|\mathbb{D}|$ grows super-exponentially in N
- **Continuous optimization:** smooth acyclicity functions $\mathcal{H}(\mathbf{A}) = 0 \iff \mathcal{D} \in \mathbb{D}$
  - $\Rightarrow$ NOTEARS [Zheng et al'18], DAGMA [Bello et al'22], CoLiDE [Saboksayr et al'24]
  - $\Rightarrow$ Scale well, but face non-convex optimization challenges
- **Order-based:** recover causal ordering, then estimate edges
  - $\Rightarrow$ TOPO [Deng et al'23], information theoretic scheme [Daskalakis et al'25]
  - $\Rightarrow$ Strong theoretical guarantees, but may neglect edge weights

Innovation in context

- Learning a DAG solely from observational data **X** is NP-hard [Chickering'96]
- **Combinatorial search:** GES [Chickering'02], PC algorithm [Spirtes et al'00]
  - ⇒ Scale poorly: $|\mathbb{D}|$ grows super-exponentially in N
- **Continuous optimization:** smooth acyclicity functions $\mathcal{H}(\mathbf{A}) = 0 \iff \mathcal{D} \in \mathbb{D}$
  - ⇒ NOTEARS [Zheng et al'18], DAGMA [Bello et al'22], CoLiDE [Saboksayr et al'24]
  - ⇒ Scale well, but face non-convex optimization challenges
- **Order-based:** recover causal ordering, then estimate edges
  - ⇒ TOPO [Deng et al'23], information theoretic scheme [Daskalakis et al'25]
  - ⇒ Strong theoretical guarantees, but may neglect edge weights

Our contribution: a deterministic, bottom-up algorithm that leverages the precision matrix structure to recover DAGs without continuous optimization or combinatorial search

Precision matrix of a linear Gaussian SEM

- Linear Gaussian SEM: $\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{z}$, with $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_N)$

  $\Rightarrow$ Signal covariance: $\boldsymbol{\Sigma} = \mathbb{E}\left[\mathbf{x}\mathbf{x}^\top\right] = \sigma^2 (\mathbf{I}_N - \mathbf{A})^{-1} (\mathbf{I}_N - \mathbf{A}^\top)^{-1}$

  $\Rightarrow$ Precision matrix: $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1} = \sigma^{-2} \left(\mathbf{I}_N - \mathbf{A} - \mathbf{A}^\top + \mathbf{A}^\top \mathbf{A}\right)$

Precision matrix of a linear Gaussian SEM

- Linear Gaussian SEM: $\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{z}$, with $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_N)$

  $\Rightarrow$ Signal covariance: $\boldsymbol{\Sigma} = \mathbb{E}\left[\mathbf{x}\mathbf{x}^\top\right] = \sigma^2 (\mathbf{I}_N - \mathbf{A})^{-1} (\mathbf{I}_N - \mathbf{A}^\top)^{-1}$

  $\Rightarrow$ Precision matrix: $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1} = \sigma^{-2} \left(\mathbf{I}_N - \mathbf{A} - \mathbf{A}^\top + \mathbf{A}^\top \mathbf{A}\right)$

Lemma 1 (Precision matrix entries). Let $\mathrm{CH}_j = \{i \in \mathcal{V} : A_{ij} \neq 0\}$ be the children of j. Then:

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

- Support of $\boldsymbol{\Theta}$ corresponds to the moralized graph of $\mathcal{D}$ [Lauritzen'96]

  $\Rightarrow$ Undirected graph, connecting all nodes in $\mathrm{PA}_i = \{j \in \mathcal{V} : A_{ij} \neq 0\}$, $i \in \mathcal{V}$

Key insight: Identifying leaf nodes and their parents

- If node i is a leaf of $\mathcal{D}$, then $\text{CH}_i = \emptyset$ and $\mathbf{a}_i = \mathbf{0}_N$

Corollary 1 (Leaf identification). A node $i \in \mathcal{V}$ is a leaf of $\mathcal{D}$ if and only if $\Theta_{ii} = \sigma^{-2}$

$\Rightarrow$ All non-leaf nodes must have $\Theta_{jj} > \sigma^{-2}$

$\Rightarrow$ Resolution gap lower bounded by $\Delta = \sigma^{-2} \min_{i,j \in \mathcal{V}} A_{ij}^2$

Key insight: Identifying leaf nodes and their parents

- If node i is a leaf of $\mathcal{D}$, then $\text{CH}_i = \emptyset$ and $\mathbf{a}_i = \mathbf{0}_N$

Corollary 1 (Leaf identification). A node $i \in \mathcal{V}$ is a leaf of $\mathcal{D}$ if and only if $\Theta_{ii} = \sigma^{-2}$

$\Rightarrow$ All non-leaf nodes must have $\Theta_{jj} > \sigma^{-2}$

$\Rightarrow$ Resolution gap lower bounded by $\Delta = \sigma^{-2} \min_{i,j \in \mathcal{V}} A_{ij}^2$

- **Upshot:** Once a leaf i is identified, we can recover its parents and edge weights:

$$A_{ij} = \begin{cases} -\sigma^2 \Theta_{ij}, & j < i \\ 0, & j \geq i \end{cases}$$

$\Rightarrow$ Off-diagonal entries in the i-th row of $\boldsymbol{\Theta}$ are scaled copies of the i-th row of $\mathbf{A}$ (i.e., PA_i)

Key insight: Identifying leaf nodes and their parents

- If node i is a leaf of $\mathcal{D}$, then $\mathrm{CH}_i = \emptyset$ and $\mathbf{a}_i = \mathbf{0}_N$

Corollary 1 (Leaf identification). A node $i \in \mathcal{V}$ is a leaf of $\mathcal{D}$ if and only if $\Theta_{ii} = \sigma^{-2}$

$\Rightarrow$ All non-leaf nodes must have $\Theta_{jj} > \sigma^{-2}$

$\Rightarrow$ Resolution gap lower bounded by $\Delta = \sigma^{-2} \min_{i,j \in \mathcal{V}} A_{ij}^2$

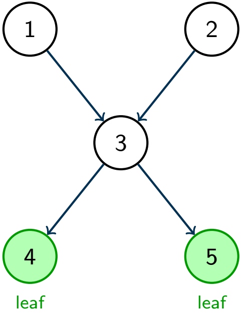
- **Upshot:** Once a leaf i is identified, we can recover its parents and edge weights:

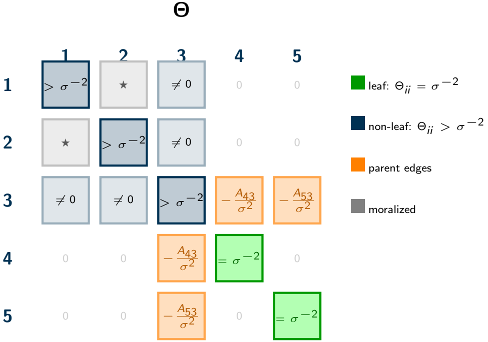
$$A_{ij} = \begin{cases} -\sigma^2 \Theta_{ij}, & j < i \\ 0, & j \geq i \end{cases}$$

$\Rightarrow$ Off-diagonal entries in the i-th row of $\boldsymbol{\Theta}$ are scaled copies of the i-th row of $\mathbf{A}$ (i.e., PA_i)

- **Core idea:** iteratively identify leaf nodes → recover their parents → prune and repeat
  - $\Rightarrow$ Bottom-up reconstruction of the entire DAG given $\boldsymbol{\Theta}$

Example: Reading the DAG from $\boldsymbol{\Theta}$





$\boldsymbol{\Theta}$

$$\begin{array}{c|ccccc} & \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \\ \hline \mathbf{1} & > \sigma^{-2} & \star & \neq 0 & 0 & 0 \\ \mathbf{2} & \star & > \sigma^{-2} & \neq 0 & 0 & 0 \\ \mathbf{3} & \neq 0 & \neq 0 & > \sigma^{-2} & -\frac{A_{43}}{\sigma^2} & -\frac{A_{53}}{\sigma^2} \\ \mathbf{4} & 0 & 0 & -\frac{A_{43}}{\sigma^2} & = \sigma^{-2} & 0 \\ \mathbf{5} & 0 & 0 & -\frac{A_{53}}{\sigma^2} & 0 & = \sigma^{-2} \end{array}$$

🟩 leaf: $\Theta_{ii} = \sigma^{-2}$

■ non-leaf: $\Theta_{ii} > \sigma^{-2}$

🟧 parent edges

▨ moralized

- Leaf rows of $\boldsymbol{\Theta}$ directly reveal parent identities and edge weights
  - $\Rightarrow$ Off-diagonal entries Θ_{ij} in leaf row i: $A_{ij} = -\sigma^2 \Theta_{ij}$
  - $\Rightarrow$ Moralized entries ($\star$) arise from shared children, not direct edges

Leaf pruning via block matrix inversion

- After identifying leaf i and recovering its incident edges, remove its contribution

  $\Rightarrow$ Partition $\mathbf{\Theta}$ with leaf i isolated

$$\mathbf{\Theta} = \begin{bmatrix} [\mathbf{\Theta}]_{ii} & [\mathbf{\Theta}]_{i,\mathcal{R}} \\ [\mathbf{\Theta}]_{\mathcal{R},i} & [\mathbf{\Theta}]_{\mathcal{R},\mathcal{R}} \end{bmatrix}$$

  $\Rightarrow$ $\mathcal{R} = \mathcal{V} \setminus \{i\}$ is the set of remaining nodes

Leaf pruning via block matrix inversion

- After identifying leaf i and recovering its incident edges, remove its contribution

  $\Rightarrow$ Partition $\boldsymbol{\Theta}$ with leaf i isolated

$$\boldsymbol{\Theta} = \begin{bmatrix} [\boldsymbol{\Theta}]_{ii} & [\boldsymbol{\Theta}]_{i,\mathcal{R}} \\ [\boldsymbol{\Theta}]_{\mathcal{R},i} & [\boldsymbol{\Theta}]_{\mathcal{R},\mathcal{R}} \end{bmatrix}$$

  $\Rightarrow$ $\mathcal{R} = \mathcal{V} \setminus \{i\}$ is the set of remaining nodes

- Updated precision matrix for the reduced DAG

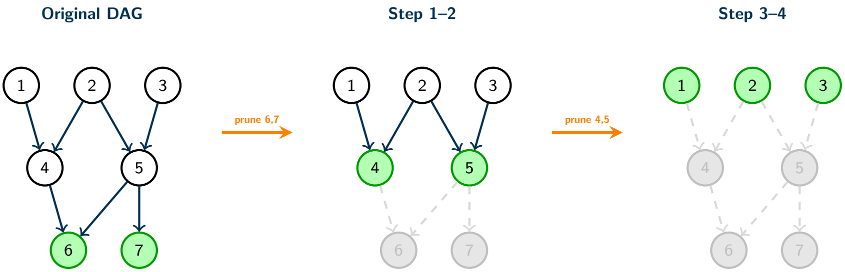
$$\boldsymbol{\Theta}_{\mathcal{R}\mathcal{R}} = [\boldsymbol{\Theta}]_{\mathcal{R},\mathcal{R}} - [\boldsymbol{\Theta}]_{ii}^{-1}[\boldsymbol{\Theta}]_{\mathcal{R},i}[\boldsymbol{\Theta}]_{i,\mathcal{R}}$$

  $\Rightarrow$ Since $[\boldsymbol{\Theta}]_{ii} = \sigma^{-2}$ for a leaf, this simplifies to a rank-one update

  $\Rightarrow$ Resulting $\boldsymbol{\Theta}_{\mathcal{R}\mathcal{R}}$ describes conditional dependencies of the pruned DAG

- Iterate: find next leaf, recover parents, prune $\rightarrow$ full DAG recovery

The BUILD algorithm



- Two phases:

  $\Rightarrow$ Phase 1: estimate $\hat{\boldsymbol{\Theta}}$ from data $\mathbf{X}$ Phase 2: iterative bottom-up DAG recovery

- Per step: find leaf $\mathcal{O}(N)$, recover parents $\mathcal{O}(d)$, prune $\mathcal{O}(d^2)$ $\Rightarrow$ Total: $\mathcal{O}(N^2)$ for Phase 2

- Guarantee: Given ensemble precision matrix $\boldsymbol{\Theta}$, BUILD exactly recovers $\mathbf{A}$

F. Kelner *et al*, "Learning some popular Gaussian graphical models without condition number bounds," *NeurIPS*, 2020

Experimental setup

- Graphs: Erdős–Rényi DAGs, $N = 200$ nodes, expected degree $d = 4$
  - $\Rightarrow$ Edge weights A_{ij} drawn uniformly from $(-2, -0.5) \cup (0.5, 2)$
- Data: $M = 1{,}000$ i.i.d. samples from linear SEM with homoscedastic Gaussian noise, $\sigma^2 = 1$
- Metrics:
  - $\Rightarrow$ SHD (Structural Hamming Distance): edge corrections to recover true graph
  - $\Rightarrow$ FDR (False Discovery Rate): fraction of spurious edges
  - $\Rightarrow$ TPR (True Positive Rate): fraction of correctly detected edges
  - $\Rightarrow$ NMSE: normalized MSE of estimated edge weights
- Baselines: CoLiDE, DAGMA, Gao et al'22, Daskalakis et al'25
- Reproducibility: Code at `https://github.com/hamedajorlou/BUILD`

M. Gao *et al*, "Optimal estimation of Gaussian DAG models," *AISTATS*, 2022

C. Daskalakis *et al*, "Learning Gaussian DAG models without condition number bounds," *ICML*, 2025

Results: BUILD versus continuous optimization methods

Table: Comparison for different refreshing rates. ER4 graphs, $N = 200$, $M = 1{,}000$. Mean ± std over 20 trials.

Baseline	SHD ↓	FDR ↓	TPR ↑	Time (s) ↓
BUILD-0.005	**17.40 ± 3.64**	**0.004 ± 0.003**	**0.983 ± 0.004**	1203.20 ± 31.65
BUILD-0.01	45.20 ± 10.89	0.035 ± 0.012	0.981 ± 0.004	620.77 ± 15.81
BUILD-0.02	81.65 ± 28.53	0.072 ± 0.029	0.977 ± 0.004	323.72 ± 7.91
BUILD-0.04	122.90 ± 34.02	0.112 ± 0.030	0.971 ± 0.007	168.17 ± 3.80
CoLiDE	114.40 ± 43.11	0.031 ± 0.026	0.888 ± 0.031	109.01 ± 22.80
DAGMA	135.95 ± 36.19	0.035 ± 0.021	0.864 ± 0.027	**93.92 ± 19.43**

- BUILD-0.005 achieves substantially lower SHD and higher TPR than all baselines

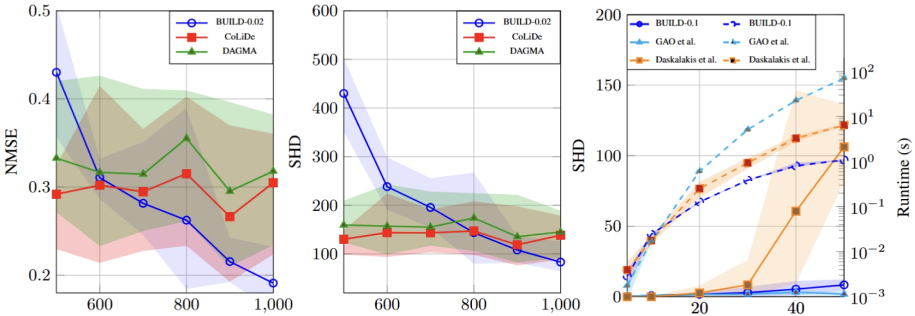
  ⇒ FDR consistently decreases with more frequent refreshing

  ⇒ Trade-off: BUILD-0.04 matches CoLiDE accuracy at comparable runtime

S. S. Saboksayr *et al*, "CoLiDE: Concomitant linear DAG estimation," *ICLR*, 2024

K. Bello *et al*, "DAGMA: Learning DAGs via M-matrices and a log-determinant acyclicity characterization," *NeurIPS*, 2022

Results: Structure recovery versus sample and graph size



(a) Number of samples (M) (b) Number of samples (M) (c) Number of nodes (N)

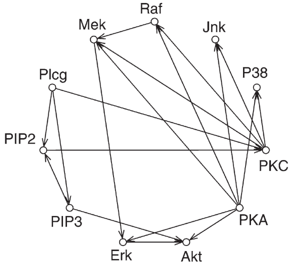
- BUILD-0.02 outperforms CoLiDE and DAGMA once $M \geq 800$ samples available

  $\Rightarrow$ Edge weight estimation (NMSE) also improves with more data

- Markedly improved error-runtime tradeoffs than order-based methods

Results: Cell-signaling data

- Tested BUILD on the Sachs dataset [Sachs et al'05]
  - $\Rightarrow$ Cytometric measurements from human immune system
  - $\Rightarrow$ Comprises $N = 11$ proteins, 17 edges, and $M = 853$ samples
  - $\Rightarrow$ Associated DAG is obtained through experimental methods



- BUILD attains lowest SHD to date for this problem

Table: DAG recovery performance on the Sachs dataset

	GOLEM-EV	GOLEM-NV	DAGMA	SortNRegress	DAGuerreotype	GES	CoLiDE-EV	CoLiDE-NV
SHD	22	15	16	13	14	13	13	**12**
SID	49	58	52	47	50	56	47	**46**
SHD-C	19	**11**	15	13	12	**11**	13	14
FDR	0.83	0.66	**0.5**	0.61	0.57	**0.5**	0.54	0.53
TPR	0.11	0.11	0.05	0.29	0.17	0.23	0.29	**0.35**

K. Sachs *et al*, "Causal protein-signaling networks derived from multiparameter single-cell data," *Science*, 2005

Summary and the road ahead

- Proposed BUILD: a deterministic, bottom-up algorithm for DAG learning
  - ⇒ Exploits precision matrix structure of linear Gaussian SEMs (Lemma 1, Corollary 1)
  - ⇒ Iteratively identifies leaf nodes, recovers parents and edge weights, and prunes
  - ⇒ Exact recovery given the ensemble precision matrix in $\mathcal{O}(N^2)$ time
  - ⇒ Refreshing scheme to mitigate finite-sample error propagation

Summary and the road ahead

- Proposed BUILD: a deterministic, bottom-up algorithm for DAG learning
  - $\Rightarrow$ Exploits precision matrix structure of linear Gaussian SEMs (Lemma 1, Corollary 1)
  - $\Rightarrow$ Iteratively identifies leaf nodes, recovers parents and edge weights, and prunes
  - $\Rightarrow$ Exact recovery given the ensemble precision matrix in $\mathcal{O}(N^2)$ time
  - $\Rightarrow$ Refreshing scheme to mitigate finite-sample error propagation
- Ongoing and future work:
  - $\Rightarrow$ Formalizing estimation error bounds and sample-complexity guarantees
  - $\Rightarrow$ Extension to non-Gaussian SEMs