TU Delft

Delft University of Technology

Document Version
Final published version

Licence
Dutch Copyright Act (Article 25fa)

Citation (APA)
Ajorlou, H., Rey , S., Mateos, G., Leus, G. J. T., & Marques, A. G. (2026). Build With Precision: Bottom-Up Inference of Linear Dags. In *Proceedings of the ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 446-450). IEEE. https://doi.org/10.1109/ICASSP55912.2026.11462602

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3315-6701-9/26/$31.00 ©2026 IEEE | DOI: 10.1109/ICASSP55912.2026.11462602

BUILD WITH PRECISION: BOTTOM-UP INFERENCE OF LINEAR DAGS

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

[1] University of Rochester, Rochester, NY, USA
[2] Universidad Rey Juan Carlos, Madrid, Spain
[3] Delft University of Technology, Delft, The Netherlands

ABSTRACT

Learning the structure of directed acyclic graphs (DAGs) from observational data is a central problem in causal discovery, statistical signal processing, and machine learning. Under a linear Gaussian structural equation model (SEM) with equal noise variances, the problem is identifiable and we show that the ensemble precision matrix of the observations exhibits a distinctive structure that facilitates DAG recovery. Exploiting this property, we propose BUILD (Bottom-Up Inference of Linear DAGs), a deterministic stepwise algorithm that identifies leaf nodes and their parents, then prunes the leaves by removing incident edges to proceed to the next step, exactly reconstructing the DAG from the true precision matrix. In practice, precision matrices must be estimated from finite data, and ill-conditioning may lead to error accumulation across BUILD steps. As a mitigation strategy, we periodically re-estimate the precision matrix (with less variables as leaves are pruned), trading off runtime for enhanced robustness. Reproducible results on challenging synthetic benchmarks demonstrate that BUILD compares favorably to state-of-the-art DAG learning algorithms, while offering an explicit handle on complexity.

***Index Terms*—** DAG structure learning; graphical model; precision matrix; topology inference; causal discovery

1. INTRODUCTION

Recovering the structure of directed acyclic graphs (DAGs) from observational data is a fundamental problem in signal processing, statistics, and machine learning, with applications in causal discovery and graphical modeling [1, 2, 3]. A common modeling framework is the linear structural equation model (SEM), where each observed variable depends linearly on its parents in the latent DAG together with exogenous noise [4, 2]. This paper addresses the structure identification problem in the specific setting of linear Gaussian SEMs with known, equal error variances. In this setting, the DAG is known to be identifiable from observational data alone [5], motivating the development of several computational methods [6, 7, 8, 9].

Related work. The problem of learning DAGs has a long history; see e.g., [5, Ch. 7.2] for a thorough treatment. Early approaches relied on purely discrete or combinatorial search methods in the space of DAGs. Noteworthy representatives include the Greedy Equivalence Search (GES) [10, 11] and constraint-based methods such as the PC algorithm [12, 13], though these approaches scale poorly due to the super-exponential growth (on the number of nodes) of the search space [14, 15]. A recent line of work develops continuous relaxations of score-based methods, where acyclicity is enforced through smooth functions amenable to gradient-based optimization. NOTEARS pioneered the trace-exponential acyclicity characterization [16], with later refinements via polynomial and log-determinant functions whose zeroth level set is the space of DAGs [17, 18, 19, **?**]. Subsequent contributions, including GOLEM [20], DAGMA, and CoLiDE [21], advanced this direction by optimizing least-squares or likelihood-based objectives augmented by acyclicity penalties, with CoLiDE further incorporating noise variance estimation for adaptivity. These methods scale well to large number of variables, but can face optimization challenges as they attemp to solve non-convex, equality-constrained problems [18]. When edge weights are nonnegative, then convex formulations are possible [22]. Here instead, we deal with general (non-sign constrained) weights and sidestep optimization issues altogether. Order-based methods recover DAGs by first learning a node ordering and then estimating edges. Recent state-of-the-art representatives include [23, 24], but these algorithms only estimate the DAG structure (neglecting edge weights, unlike ours), albeit with strong theoretical support recovery guarantees.

More germane to our approach in this paper, another way to tackle DAG learning is to restrict the search space to a superstructure, namely an undirected skeleton encompassing a restricted class of possible DAGs. For Gaussian data, the precision (i.e., inverse covariance) matrix of the observations encodes conditional independencies and has well-documented links to the so-termed moralized graph associated to the DAG [2, 5, 25]; see also [26] for an influential work exploring general non-Gaussian settings. This creates a strong incentive to leverage high-quality precision matrix estimators, such as graphical Lasso [27], as a precursor to DAG structure inference. This is the path we follow here, but cautiously, since we show that for linear SEMs the resulting precision matrix can be severely ill-conditioned. Interestingly, GreedyPrune avoids restrictive condition number assumptions and provides fixed polynomial-time guarantees for recovering the precision matrix in certain classes of Gaussian graphical models [28]. Motivated by these insights on the role of precision matrices to unveil latent DAG structure, we next present our own approach that builds directly on this valuable connection.

Contributions. We propose BUILD (Bottom-Up Inference of Linear DAGs), a deterministic algorithm that iteratively identifies leaf nodes and their parents, then prunes the leaves by removing incident edges to proceed to the next step, exactly reconstructing the DAG from the ensemble precision matrix of a linear Gaussian SEM model with equal noise variances. The algorithm is motivated by the favorable structure we reveal in the population-level precision matrix (see Lemma 1 and Corollary 1). In practice, we only have data to work with and hence an imperfect precision matrix estimate. A key feature of BUILD is an error mitigation strategy that balances accuracy and

Work in this paper was supported by the NSF award ECCS-2231036, the Spanish AEI (10.13039/501100011033) grant PID2022-136887NB-I00, and the Community of Madrid via the Ellis Madrid Unit and grants URJC/CAM F1180 and TEC-2024/COM-89.

Authorized licensed use limited to: TU Delft Library. Downloaded on May 12,2026 at 07:13:24 UTC from IEEE Xplore. Restrictions apply.

runtime by periodically re-estimating the precision matrix, as leaves are pruned and the problem dimensionality decreases. Through extensive synthetic experiments, BUILD demonstrates strong performance across multiple evaluation metrics, achieving accurate edge detection and weight estimation while maintaining competitive runtime. Moreover, its performance scales favorably as the number of nodes increases. To ensure transparency and reproducibility, we publicly share the implementation of BUILD alongside the paper.

2. PRELIMINARIES AND PROBLEM STATEMENT

We introduce the required background on DAGs and linear SEMs needed to formally state the DAG topology inference problem.

Directed acyclic graphs. Let $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ denote a DAG, where $\mathcal{V} = \{1, \dots, N\}$ is the set nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of directed edges. The convention we adopt is that $(i, j) \in \mathcal{E}$ means that there is an arc $j \to i$. Because $\mathcal{D}$ contains no directed cycles or self loops, $\mathcal{V}$ admits a (generally non-unique) topological ordering whereby $(i, j) \in \mathcal{E}$ implies that node j precedes node i. Thus, every DAG defines a unique partial order over the set $\mathcal{V}$, where $j < i$ if node j is a *predecessor* of i, meaning there exists a directed path from j to i. The weighted adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ encodes the connectivity structure of $\mathcal{D}$, with $A_{ij} \neq 0$ if and only if $(i, j) \in \mathcal{E}$. When $\mathcal{V}$ is ordered topologically, $\mathbf{A}$ is strictly lower triangular.

Linear structural equation models. Let us assume $\mathcal{D}$ captures conditional independencies among the variables in the random vector $\mathbf{x} = [x_1, \dots, x_N]^\top \in \mathbb{R}^N$. If the joint distribution $\mathbb{P}(\mathbf{x})$ satisfies a Markov property over $\mathcal{D}$, it implies that each random variable x_i is solely dependent on its parents $\text{PA}_i = \{j \in \mathcal{V} : A_{ij} \neq 0\}$ [2]. This work focuses on *linear* SEMs to generate such a probability distribution, where the relationship between each random variable and its parents is expressed as $x_i = \sum_{j \in \text{PA}_i} A_{ij} x_j + z_i$, $\forall i \in \mathcal{V}$, where $\mathbf{z} = [z_1, \dots, z_N]^\top$ is a zero-mean Gaussian vector of mutually independent, exogenous noises with known variance σ^2. For a dataset $\mathbf{X} \in \mathbb{R}^{N \times M}$ consisting of M i.i.d. samples drawn from $\mathbb{P}(\mathbf{x})$, the linear SEM can be expressed in matrix form as $\mathbf{X} = \mathbf{A}\mathbf{X} + \mathbf{Z}$.

Problem statement. Given the data matrix $\mathbf{X}$ generated by a linear SEM, the goal is to recover the underlying DAG $\mathcal{D}$ by estimating its adjacency matrix $\mathbf{A}$. Under the assumption of Gaussian $\mathbf{z}$ with equal variances, then $\mathcal{D}$ is identifiable from the joint distribution of $\mathbf{x}$ alone (i.e., from observational data); see e.g., [2, Prop. 7.5].

3. BUILD: BOTTOM-UP INFERENCE OF LINEAR DAGS

Here, we present the proposed DAG inference algorithm. For a linear Gaussian SEM with known equal noise variances, we first describe how the adjacency matrix $\mathbf{A}$ of $\mathcal{D}$ can be reconstructed from the precision matrix $\mathbf{\Theta}$ of $\mathbf{x}$. We follow a bottom-up approach that sequentially identifies and prunes leaf nodes and incident edges from their parents. We then touch upon important implementation details, namely ill-conditioned precision matrix estimation and mitigation of finite-sample induced errors in the DAG reconstruction process.

DAG recovery from the ensemble precision matrix. Recall the linear SEM $\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{z}$, where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_N)$. The signal covariance matrix is given by $\mathbf{\Sigma} := \mathbb{E}\left[\mathbf{x}\mathbf{x}^\top\right] = \sigma^2 (\mathbf{I}_N - \mathbf{A})^{-1} (\mathbf{I}_N - \mathbf{A}^\top)^{-1}$ and, hence, the precision matrix $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$ has the form

$$\begin{aligned} \mathbf{\Theta} &= \sigma^{-2} \left(\mathbf{I}_N - \mathbf{A}^\top\right) \left(\mathbf{I}_N - \mathbf{A}\right) \\ &= \sigma^{-2} \left(\mathbf{I}_N - \mathbf{A} - \mathbf{A}^\top + \mathbf{A}^\top \mathbf{A}\right). \end{aligned} \tag{1}$$

The following simple result characterizes the entries of the precision matrix $\mathbf{\Theta}$, and will be central to our algorithmic approach.

Lemma 1 (Precision matrix entries)**.** Let $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_N] \in \mathbb{R}^{N \times N}$ be the adjacency matrix of $\mathcal{D}$ and write $\text{supp}(\mathbf{a}_j) \equiv \text{CH}_j := \{i \in \mathcal{V} : A_{ij} \neq 0\}$, the children of $j \in \mathcal{V}$. The (scaled by σ^2) entries of the (symmetric) precision matrix $\mathbf{\Theta}$ in (1) are given by

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

Proof. Follows immediately by noticing $\text{diag}(\mathbf{A}) = \mathbf{0}_N$ because a DAG has no self loops, and from $[\mathbf{A}^\top \mathbf{A}]_{ij} = \mathbf{a}_i^\top \mathbf{a}_j$. □

The support of $\mathbf{\Theta}$ corresponds to the so-termed moralized graph of $\mathcal{D}$ [26], an undirected graph obtained by connecting all nodes within each parent set PA_i and also dropping the directionality of all edges in $\mathcal{E}$. Now, if node i is a leaf of $\mathcal{D}$ then $\text{CH}_i = \emptyset$ and $\mathbf{a}_i = \mathbf{0}_N$. This observation along with Lemma 1 suggests leaf nodes can be unequivocally identified from the diagonal entries of $\mathbf{\Theta}$.

Corollary 1. *A node $i \in \mathcal{V}$ is a leaf of $\mathcal{D}$ if and only if $\Theta_{ii} = \sigma^{-2}$. All other non-leaf nodes have $\Theta_{jj} > \sigma^{-2}$ and the gap or resolution limit is lower bounded by $\Delta = \sigma^{-2} \min_{i,j \in \mathcal{V}} A_{ij}^2$.*

Moreover, it follows from (2) that we can also identify the parents PA_i of leaf i and the edge weights $A_{ij} \neq 0$ (recall σ^2 is known). This is because $\text{CH}_i = \emptyset$, hence the off-diagonal entries in the i-th row (and column) of $\mathbf{\Theta}$ are scaled (by $-\sigma^{-2}$) copies of the i-th row of $\mathbf{A}$, whose support is precisely PA_i. All in all, if we are given $\mathbf{\Theta}$ and identify $i \in \mathcal{V}$ as a leaf (cf. Corolloray 1), then we can recover the i-th row of the adjacency matrix as

$$A_{ij} = \begin{cases} -\sigma^2 \Theta_{ij}, & j < i \\ 0, & j \geq i \end{cases}. \tag{3}$$

Of course, we also know the i-th column of $\mathbf{A}$ is $\mathbf{a}_i = \mathbf{0}_N$. This observation forms the key insight behind the proposed DAG recovery method, at least in the idealized setting where the ensemble precision matrix $\mathbf{\Theta}$ is available. The idea is to iteratively identify leaf nodes using Corollary 1, recover their parents from $\mathbf{\Theta}$ using (3), and then remove the contribution of each leaf (the node itself and its incident edges) using (2). The resulting matrix describes the conditional dependence relations of the reduced DAG. Repeating this stepwise bottom-up process enables full recovery of the underlying DAG.

The two last steps of the aforementioned process can be related to eliminating a leaf node from the data and recomputing the precision matrix over the leftover nodes. To clarify this relation, let us split the precision (and covariance) matrix in 4 blocks (for visual convenience we permute rows and columns so that leaf i is depicted first) and, a block matrix inversion identity yields

$$\mathbf{\Theta} = \left[\begin{array}{c|c} [\mathbf{\Theta}]_{ii} & [\mathbf{\Theta}]_{i,\mathcal{R}} \\ \hline [\mathbf{\Theta}]_{\mathcal{R},i} & [\mathbf{\Theta}]_{\mathcal{R},\mathcal{R}} \end{array}\right], \tag{4}$$

$$\mathbf{\Theta}_{\mathcal{R}\mathcal{R}} = [\mathbf{\Theta}]_{\mathcal{R},\mathcal{R}} - [\mathbf{\Theta}]_{ii}^{-1} [\mathbf{\Theta}]_{\mathcal{R},i} [\mathbf{\Theta}]_{i,\mathcal{R}}, \tag{5}$$

where $\mathcal{R} = \mathcal{V} \setminus \{i\}$ is the set of remaining nodes, $[\mathbf{\Theta}]_{ii} = \Theta_{ii} = \sigma^{-2}$ is the scalar diagonal entry corresponding to the selected leaf i, $[\mathbf{\Theta}]_{i,\mathcal{R}} \in \mathbb{R}^{1 \times |\mathcal{R}|}$ is the row vector with entries $-\sigma^{-2} A_{ij}$, $j \in \mathcal{R}$, and $[\mathbf{\Theta}]_{\mathcal{R}\mathcal{R}} \in \mathbb{R}^{|\mathcal{R}| \times |\mathcal{R}|}$ is the precision submatrix over the leftover nodes (prior to pruning). After removing i, the precision matrix for

Authorized licensed use limited to: TU Delft Library. Downloaded on May 12,2026 at 07:13:24 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 BUILD: Bottom-Up Inference of Linear DAGs

Input: $\hat{\mathbf{\Theta}}$, $\mathbf{X}$, σ^2, threshold ε, refresh rate $\rho \in [0,1]$
Output: $\widehat{\mathbf{A}} \in \mathbb{R}^{N\times N}$

$\mathbf{A} \leftarrow \mathbf{0}_{N\times N}$, $\tau \leftarrow 0$, $\mathbf{\Theta} \leftarrow \hat{\mathbf{\Theta}}$, $\mathcal{Q} \leftarrow \emptyset$ ▷ pruned leaves in $\mathcal{Q}$
$\mathcal{T} \leftarrow \{\lfloor t\rho N \rfloor : t = 1, 2, \ldots, N-1\}$ ▷ refresh checkpoints
while $|\mathcal{Q}| < N$ **do**
  $\mathcal{R} \leftarrow \{1, \ldots, N\} \setminus \mathcal{Q}$ ▷ remaining nodes
  if $\tau \in \mathcal{T}$ **then** ▷ time to re-estimate $\mathbf{\Theta}_{\mathcal{R}\mathcal{R}}$
    $\hat{\mathbf{\Theta}}_{\mathcal{R}\mathcal{R}} \leftarrow \textsc{GreedyPrune}(\mathbf{X}_{\mathcal{R}})$
    $[\mathbf{\Theta}]_{\mathcal{R},\mathcal{R}} \leftarrow \hat{\mathbf{\Theta}}_{\mathcal{R}\mathcal{R}}$
  end if
  $i \leftarrow \arg\min_{u \notin \mathcal{Q}} \{\Theta_{uu} \geq \varepsilon\}$ **break if none** ▷ find leaf
  $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{i\}$, $\tau \leftarrow \tau + 1$
  $\mathbf{a} \leftarrow -\sigma^2 \mathbf{\Theta}_{i,:}$; set $a_i \leftarrow 0$; $\forall j \neq i$, set $a_j \leftarrow 0$ if $|a_j| < \varepsilon$
  $\mathbf{A}_{i,:} \leftarrow \mathbf{a}$ ▷ connectivity of i to its parents [cf. (3)]
  for each j with $|a_j| \geq \varepsilon$ **do**
    $\mathbf{\Theta}_{j,:} \leftarrow \mathbf{\Theta}_{j,:} - \sigma^{-2} a_j \mathbf{a}$ ▷ effect of i on $\mathcal{R}$ [cf. (5)]
  end for
  $\mathbf{\Theta}_{i,:} \leftarrow \mathbf{0}$, $\mathbf{\Theta}_{:,i} \leftarrow \mathbf{0}$ ▷ prune leaf node i
end while
return $\widehat{\mathbf{A}} = \mathbf{A}$

the next step is $\mathbf{\Theta}_{\mathcal{R}\mathcal{R}} = [\mathbf{\Theta}]_{\mathcal{R},\mathcal{R}} - [\mathbf{\Theta}]_{ii}^{-1} [\mathbf{\Theta}]_{\mathcal{R},i} [\mathbf{\Theta}]_{i,\mathcal{R}}$. If i is indeed a leaf node, the elimination of the i-th row and column and the update of the remaining entries via (5) is similar to the process described in the previous paragraph.

Building on this core idea, we now introduce our proposed approach, BUILD (Bottom-Up Inference of Linear DAGs), which operates in two phases. Given data $\mathbf{X} \in \mathbb{R}^{N\times M}$, in the first phase we estimate the precision matrix borrowing existing methods from the literature. In the second phase (our main contribution), we apply Algorithm 1 to recover the underlying DAG structure from $\hat{\mathbf{\Theta}}$.

Implementation details. With access to the ensemble precision matrix $\mathbf{\Theta}$, BUILD deterministically recovers the adjacency matrix $\mathbf{A}$ in $\mathcal{O}(N^2)$ time. In practice, however, the population-level precision matrix is unknown and it must be inferred from observational data $\mathbf{X}$. This can be particularly challenging for high-dimensional linear SEMs with equal noise variance σ^2, as the covariance matrix $\mathbf{\Sigma} = \sigma^2(\mathbf{I}_N - \mathbf{A})^{-1}(\mathbf{I}_N - \mathbf{A}^\top)^{-1}$ (and hence $\mathbf{\Theta}$) will be badly conditioned. Indeed, the Neumann expansion $(\mathbf{I}_N - \mathbf{A})^{-1} = \sum_{k=0}^{N-1} \mathbf{A}^k$ reveals that variances at downstream nodes in $\mathcal{D}$ accumulate noise contributions from all their ancestors. This noise accumulation effect leads to large disparities in the marginal variances of $x_1, \ldots, x_N$, and thus in the eigenvalues of $\mathbf{\Sigma}$. The result is an ill-conditioned precision matrix, a phenomenon particularly severe in DAGs with long dependency chains, high branching, or heterogeneous edge weights. For instance, in a chain $1 \to 2 \to \cdots \to N$ with edge weights $A_{i,i+1} = k > 1$, the condition number of $\mathbf{\Theta}$ grows as k^{2N}. We employ GreedyPrune [28] to estimate the precision matrix in the initial phase, which we found to perform best in ill-conditioned settings where e.g., graphical Lasso fails miserably.

We now provide a more detailed algorithmic description of BUILD. We examine how the finite sample-induced errors in $\hat{\mathbf{\Theta}}$ affect the estimation of $\mathbf{A}$, and describe our mitigation strategies. At each iteration, we restrict attention to the active set of variables $\mathcal{R}$. We designate as a leaf the node $i \in \mathcal{R}$ whose diagonal entry in the precision matrix attains the smallest value above a fixed tolerance ϵ, thereby avoiding spurious numerical artifacts leading to leaf misidentification. We then recover its parent relationships from the corresponding row of the precision matrix ($\mathbf{\Theta}_{i,:}$ using Matlab notation), discarding entries with negligible magnitude to suppress the effect of estimation noise. The precision matrix is then updated by removing the contribution of this node (leaf pruning via block matrix inversion described earlier), and the process continues on the reduced system, which has the same structure as before. Repeating this procedure until no nodes remain yields $\hat{\mathbf{A}}$.

As mentioned earlier, inaccurately estimated edges may fail to completely remove the influence of discovered leaf nodes, leaving residual effects in the updated precision matrix. For large DAGs, these residuals propagate further, compounding over iterations and leading to a snowball effect that may result in catastrophic edge detection errors. To mitigate this phenomenon, we introduce a refreshing scheme in which the precision matrix is re-estimated at fixed intervals. This allows BUILD to operate on an updated precision matrix (with fewer variables as leafs are pruned, thus easier to estimate), resetting past accumulated errors. Algorithm 1 considers only the active nodes $\mathcal{R}$ that have not yet been processed, extracts the corresponding rows and columns of the sample covariance matrix, and re-estimates the precision matrix using GreedyPrune as if operating on a newly pruned graph. We find that this scheme markedly mitigates the effects of error accumulation. In the experiments, we test several variants of BUILD with different refresh rates to provide further insight into the performance versus complexity tradeoffs.

Closing remarks. The main bottleneck of our approach is its sensitivity to errors in $\hat{\mathbf{\Theta}}$. The quality of $\hat{\mathbf{\Theta}}$ directly affects BUILD's ability to recover $\mathbf{A}$. With more data, the first phase yields a more reliable precision matrix, and downstream DAG estimation improves accordingly. Also, re-estimation of $\hat{\mathbf{\Theta}}$ offers added robustness at the price of increased runtime. We examine these tensions empirically in the numerical tests that follow.

Table 1: Comparison of continuous optimization methods with variants of BUILD under different refreshing rates. Metrics reported as mean $\pm$ std over 20 trials. Best entries are highlighted in **bold**.

Baseline	SHD ↓	FDR ↓	TPR ↑	Time (s) ↓
BUILD–0.005	$\mathbf{17.40 \pm 3.64}$	$\mathbf{0.004 \pm 0.003}$	$\mathbf{0.983 \pm 0.004}$	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	$\mathbf{93.92 \pm 19.43}$

4. NUMERICAL TESTS

We evaluate the performance of BUILD[1] by benchmarking it against recent state-of-the-art DAG structure learning algorithms, including CoLiDE [21], DAGMA [19], Gao et al. [23], and Daskalakis et al. [24]. Unless otherwise stated, the synthetic data are generated according to a linear SEM with homoscedastic Gaussian noise with $\sigma^2 = 1$. For the experiments, we consider Erdős–Rényi DAGs with $N = 200$ nodes and expected degree $d = 4$. The sample size is $M = 1{,}000$ observations. Edge weights A_{ij} are drawn uniformly from the range $(-2, -0.5) \cup (0.5, 2)$. This is a standard and challenging setting in the DAG estimation literature [21, 19]. The task is to recover the ground-truth DAG, and performance is evaluated in terms of structural Hamming distance (SHD), true and false positive rates (TPR and FDR), and runtime. Comparative results are reported in Table 1. Mean $\pm$ standard deviation over 20 trials is reported.

[1]Code at `https://github.com/hamedajorlou/BUILD`

Authorized licensed use limited to: TU Delft Library. Downloaded on May 12,2026 at 07:13:24 UTC from IEEE Xplore. Restrictions apply.

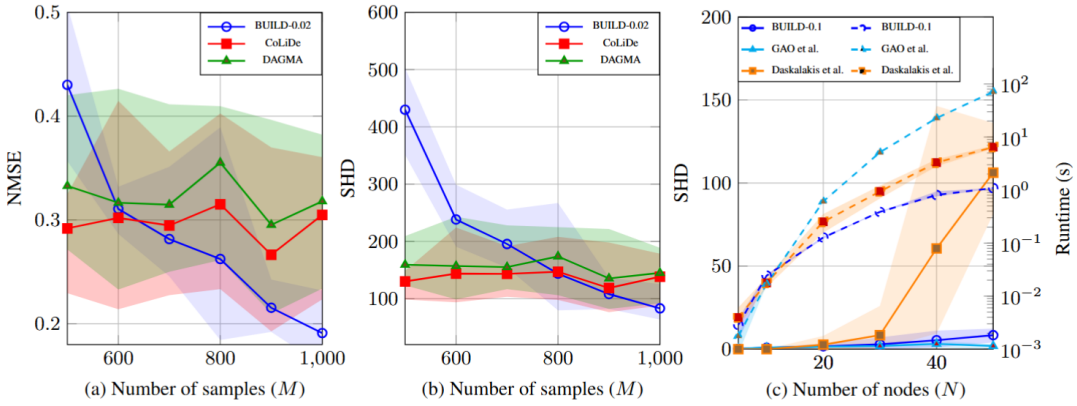


Fig. 1: Performance of BUILD compared with state-of-the-art baselines. Results are averaged over 10 trials, with shaded regions indicating the 10th and 90th percentiles. (a) NMSE of edge estimation as a function of sample size. (b) SHD as a function of sample size. (c) Comparison with order-based methods: SHD (left) and runtime in log scale (right) versus the number of nodes.

Discussion and findings. The variants of BUILD reported in Table 1 correspond to different precision matrix refresh rates. For instance, BUILD-0.1 means $\mathbf{\Theta}$ is re-estimated whenever 10% of the nodes have been pruned during the bottom-up DAG construction process.

In principle, BUILD can exactly recover the true underlying DAG if provided with the ensemble precision matrix. However, in practice the precision matrix is only available through empirical estimation, which inevitably introduces errors. As BUILD proceeds by sequentially removing the influence of leaf nodes, these estimation errors propagate and accumulate, ultimately leading to spurious edge detections. The key insight is that BUILD's accuracy is highly sensitive to precision matrix estimation error. Refreshing the precision matrix mitigates this issue by discarding the accumulated error and recalibrating the algorithm with a better estimate, thereby improving edge recovery. This effect is most evident in the FDR reported in Table 1. As the refresh rate increases, the FDR consistently decreases. Naturally, each refresh incurs additional computational cost, creating a trade-off between accuracy and runtime. As an extreme case for a DAG with $N = 200$ nodes, we run BUILD-0.005, which refreshes the precision matrix after processing each node. While admittedly computationally expensive, this variant achieves substantially higher accuracy than competing continuous optimization methods, highlighting the merits of error correction through re-estimation of $\hat{\mathbf{\Theta}}$.

Edge weight estimation. To assess edge-weight estimation performance not captured by the metrics in Table 1, we compute the Normalized MSE (NMSE) between the estimated adjacency matrix $\widehat{\mathbf{A}}$ and the ground truth $\mathbf{A}^\star$, defined as $\text{NMSE} = \|\widehat{\mathbf{A}} - \mathbf{A}^\star\|_F^2 / \|\mathbf{A}^\star\|_F^2$, across different sample sizes; see Fig. 1(a). In BUILD-0.02, the precision matrix is refreshed every 2% of the nodes processed, which for $N = 200$ nodes amounts to every 4 nodes. As M increases, the precision matrix estimation error decreases, leading to more accurate edge recovery in the second phase of the algorithm. This improved accuracy allows BUILD-0.02 to outperform the other two baselines, at comparable complexity (see Table 1).

Structure recovery. In many applications, the primary objective is to recover the correct support of $\mathbf{A}$. With this in mind, in Fig. 1(b), we report the SHD of the estimated DAGs as a function of the number of samples M. As the sample size grows, the error in precision matrix estimation decreases, which leads to more reliable support recovery. In this setting, BUILD-0.02 achieves the lowest SHD and outperforms the other models once 800 samples are available.

Comparison with order-based methods. The previous experiments compared BUILD against continuous optimization methods. We now turn our attention to a different line of order-based approaches with promising performance, namely Gao et al. [23] and Daskalakis et al. [24]. Unlike optimization-based methods, these algorithms focus on support recovery rather than estimating edge weights, which is an inherent limitation. For this reason, we present their results separately in Fig. 1(c), where the left panel shows SHD and the right panel reports runtime (in log-scale). Dashed lines correspond to the runtime of each baseline. We observe that Gao et al. [23] achieves consistently low error as the number of nodes grows, but its runtime grows rapidly, making it impractical for larger networks. By contrast, Daskalakis et al. [24] is more efficient but suffers from high error in this edge weight regime, even though it nearly recovers the ground truth when weights are restricted to $(-1, -0.5) \cup (0.5, 1)$. In terms of SHD, BUILD-0.1 matches the accuracy of Gao et al. [23] while remaining scalable, and it is consistently faster than both baselines. This makes BUILD a viable choice for large-scale DAG learning, where the alternatives either become computationally prohibitive or incur significant error.

5. CONCLUSIONS, LIMITATIONS, AND FUTURE WORK

We introduced BUILD, a deterministic algorithm for DAG structure identification using the observations' precision matrix. BUILD iteratively identifies and prunes leave nodes, reconstructing the DAG in a bottom-up fashion. To keep finite sample-induced error propagation in check, optional re-estimation of the precision matrix is considered. On synthetic linear Gaussian SEM data, BUILD achieved lower SHD and higher TPR than representative baselines while remaining computationally competitive. A limitation of the algorithm is its sensitivity to the accuracy of the estimated precision matrix, especially given that the problems tend to be badly conditioned. Moreover, refreshing the weights discards many previously computed entries of the precision matrix that are already contaminated by error, which introduces inefficiency. These limitations point toward natural directions for future research, including concomitant estimation of the noise variance and DAG structure, formalizing sample-complexity guarantees while neglecting the refresh mechanism for analytical tractability, designing adaptive refresh schedules to balance accuracy and efficiency, and extending the approach to non-Gaussian and nonlinear SEMs. Exploring structure estimation in the presence of interventional data is also of interest.

Authorized licensed use limited to: TU Delft Library. Downloaded on May 12,2026 at 07:13:24 UTC from IEEE Xplore. Restrictions apply.

6. REFERENCES

[1] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[2] J. Peters, D. Janzing, and B. Schölkopf, *Elements of Causal Inference: Foundations and Learning Algorithms*, MIT Press, 2017.

[3] F. Xia, K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, and H. Liu, "Graph learning: A survey," *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 2, pp. 109–127, 2021.

[4] S. Shimizu, P. O. Hoyer, A. Hyvärinen, and A. Kerminen, "A linear non-gaussian acyclic model for causal discovery," *J. Mach. Learning Res.*, vol. 7, no. 72, pp. 2003–2030, 2006.

[5] J. Peters and P. Bühlmann, "Identifiability of gaussian structural equation models with equal error variances," *Biometrika*, vol. 101, no. 1, pp. 219–228, 2014.

[6] W. Chen, M. Drton, and Y. S. Wang, "On causal discovery with an equal-variance assumption," *Biometrika*, vol. 106, no. 4, pp. 973–980, Sep. 2019.

[7] M. Gao, Y. Ding, and B. Aragam, "A polynomial-time algorithm for learning nonparametric causal graphs," in *Conf. Neural Inform. Process. Syst.*, 2020.

[8] A. Ghoshal and J. Honorio, "Learning identifiable Gaussian Bayesian networks in polynomial time and sample complexity," in *Conf. Neural Inform. Process. Syst.*, 2017, pp. 6457–6466.

[9] A. Ghoshal and J. Honorio, "Information-theoretic limits of Bayesian network structure learning," in *Int. Conf. Artificial Intell., Statist.*, A. Singh and J. Zhu, Eds., Fort Lauderdale, FL, USA, 20–22 Apr 2017, vol. 54 of *Proceedings of Machine Learning Research*, pp. 767–775, PMLR.

[10] D. M. Chickering, "Optimal structure identification with greedy search," *J. Mach. Learning Res.*, vol. 3, pp. 507–554, Nov. 2002.

[11] R. Sanchez-Romero J. Ramsey, M. Glymour and C. Glymour, "A million variables and more: The fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images," *International Journal of Data Science and Analytics*, vol. 3, pp. 121–129, 2017.

[12] C. Glymour P. Spirtes and R. Scheines, *Causation, Prediction, and Search*, vol. 81, The MIT Press, 2000.

[13] J. Peters P. Bühlmann and J. Ernest, "CAM: Causal additive models, high-dimensional order search and penalized regression," *Annals of Statistics*, vol. 42, no. 6, pp. 2526–2556, 2014.

[14] D. M. Chickering, "Learning Bayesian networks is NP-complete," *Learning from Data: Artificial Intelligence and Statistics V*, pp. 121–130, 1996.

[15] D. M. Chickering, D. Heckerman, and C. Meek, "Large-sample learning of Bayesian networks is NP-hard," 2004, pp. 151–158.

[16] X. Zheng, B. Aragam, P. Ravikumar, and E. Xing, "DAGs with no tears: Continuous optimization for structure learning," in *Conf. Neural Inform. Process. Syst.*, 2018, pp. 9472–9483.

[17] T. Gao Y. Yu, J. Chen and Mo Yu, "DAG-GNN: DAG structure learning with graph neural networks," in *Int. Conf. Mach. Learning*. 2019, pp. 7154–7163, PMLR.

[18] D. Wei, T. Gao, and Y. Yu, "DAGs with no fears: A closer look at continuous optimization for learning Bayesian networks," in *Conf. Neural Inform. Process. Syst.*, 2020, vol. 33, pp. 3895–3906.

[19] K. Bello, B. Aragam, and P. Ravikumar, "DAGMA: Learning DAGs via M-matrices and a log-determinant acyclicity characterization," in *Conf. Neural Inform. Process. Syst.*, 2022, vol. 35, pp. 8226–8239.

[20] I. Ng, C. Squires, and A. Anandkumar, "Golem: Generative models and structure learning without acyclicity constraints," in *Int. Conf. Mach. Learning*.

[21] S. S. Saboksayr, G. Mateos, and M. Tepper, "ColiDE: Concomitant linear DAG estimation," in *Int. Conf. Learn. Representations*, 2024.

[22] S. Rey, S. S. Saboksayr, and G. Mateos, "Non-negative weighted DAG structure learning," in *IEEE Int. Conf. Acoust., Speech and Signal Process.*, 2025, pp. 1–5.

[23] M. Gao, W. M. Tai, and B. Aragam, "Optimal estimation of gaussian dag models," *arXiv preprint arXiv:2201.10548*, 2022.

[24] C. Daskalakis, V. Kandiros, and R. Yao, "Learning gaussian dag models without condition number bounds," in *Int. Conf. Mach. Learning*, 2025.

[25] S. L. Lauritzen, *Graphical Models*, Oxford University Press, 1996.

[26] P. Loh and P. Bühlmann, "High-dimensional learning of linear causal networks via inverse covariance estimation," in *Int. Conf. Mach. Learning*, 2014, pp. 1300–1308.

[27] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[28] J. Kelner, F. Koehler, R. Meka, and A. Moitra, "Learning some popular Gaussian graphical models without condition number bounds," *Conf. Neural Inform. Process. Syst.*, pp. 1236–1249, 2020.

Authorized licensed use limited to: TU Delft Library. Downloaded on May 12,2026 at 07:13:24 UTC from IEEE Xplore. Restrictions apply.