Scalable Probabilistic Causal Structure Discovery

Dhanya Sridhar\textsuperscript{1}, Jay Pujara\textsuperscript{2}, Lise Getoor\textsuperscript{1},
\textsuperscript{1} University of California Santa Cruz
\textsuperscript{2} University of Southern California
\{dsridhar, getoor\}@soe.ucsc.edu, jay@cs.umd.edu

\section*{Abstract}

Complex causal networks underlie many real-world problems, from the regulatory interactions between genes to the environmental patterns used to understand climate change. Computational methods seek to infer these casual networks using observational data and domain knowledge. In this paper, we identify three key requirements for inferring the structure of causal networks for scientific discovery: (1) robustness to noise in observed measurements; (2) scalability to handle hundreds of variables; and (3) flexibility to encode domain knowledge and other structural constraints. We first formalize the problem of joint probabilistic causal structure discovery. We develop an approach using probabilistic soft logic (PSL) that exploits multiple statistical tests, supports efficient optimization over hundreds of variables, and can easily incorporate structural constraints, including imperfect domain knowledge. We compare our method against multiple well-studied approaches on biological and synthetic datasets, showing improvements of up to 20\% in F1-score over the best performing baseline in realistic settings.

\section{Introduction}

The problem of causal structure discovery (CSD) consists of inferring a network of cause-and-effect relationships between many variables using observational data and domain knowledge. In contrast to the estimation of single causal relationships, CSD finds consistent causal graphs over all variables, exponentially increasing problem complexity. CSD is an important task for facilitating scientific discovery, such as determining regulatory networks amongst genes [Friedman, 2004; Liu et al., 2016] and understanding influences between atmospheric patterns to better forecast climate events [Ebert-Uphoff and Deng, 2012].

Computational methods for causal structure discovery face several critical challenges. First, observational data is frequently noisy, containing spurious correlations between variables. Second, even with simplifying assumptions, CSD requires searching over exponentially many potential causal graphs, posing a scalability bottleneck. Finally, CSD requires incorporating heterogeneous domain knowledge of differing reliabilities, such as ontological and experimental evidence. Thus, successful CSD approaches must be robust, scalable, and flexible to succeed on real-world problems.

Existing methods for CSD have largely been evaluated in synthetic and low-noise settings that do not accurately represent the challenges of real-world domains. Traditional CSD approaches make locally greedy and iterative decisions, improving scalability at the cost of robustness. However, recent approaches based on logical satisfiability (SAT) [Magliacane et al., 2016] or linear programming (LP) [Cussens, 2011] have shown the benefits of enforcing global constraints on the causal graph structure through joint inference.

In this paper, we extend the joint inference view and propose a novel approach, \textsc{CausPSL}, that provides an attractive compromise between robustness to noise, scalability, and flexibility. We explore these trade-offs through extensive experimental evaluation on biological datasets, demonstrating significant performance gains on both real-world data and synthetic benchmarks. We formulate CSD as an inference problem by defining a joint probability distribution over causal graphs. Our approach defines this distribution by unifying constraints from statistical tests, side information, and domain knowledge. We implement \textsc{CausPSL} using the probabilistic soft logic (PSL) framework [Bach et al., 2017] which defines a hinge-loss Markov random field and supports efficient MAP inference. In experiments, we demonstrate several key strengths of \textsc{CausPSL}:

- \textbf{Robustness via Redundancy}: \textsc{CausPSL} exploits redundancy by using multiple statistical tests and soft constraints, mitigating noisy inputs.
- \textbf{Efficient Performance}: \textsc{CausPSL} scales to causal graphs with hundreds of variables via exact and efficient MAP inference.
- \textbf{Flexible Modeling}: \textsc{CausPSL} encodes both well-studied structural constraints and novel long- and short-range constraints with an easily extensible logical syntax.

We validate the features of \textsc{CausPSL} on realistic experimental settings including gene regulatory networks and protein signaling datasets, showing increases in F1-score of up to 20\% over state-of-the-art CSD methods.
2 Background and Related Work

Approaches to CSD fall into two major categories: score- and constraint-based. Our work extends constraint-based methods. Traditional constraint-based procedures, most notably the PC algorithm [Spirtes and Glymour, 1991], start from complete undirected graphs and iteratively prune edges between variables that are independent. These methods then maximally orient the remaining edges with rules that enforce faithfulness between observed conditional independences and graph d-separation constraints. Despite soundness and completeness guarantees under perfect inputs [Spirtes and Glymour, 1991; Zhang, 2008; Colombo and Maathuis, 2014; Ramsey et al., 2006], performance of these methods suffers under the imperfect data conditions most commonly present in real problems. Our work is motivated by recent approaches that cast CSD as a SAT instance, using off-the-shelf solvers to find causal graphs that minimally violate multiple, conflicting independence statements weighted by their confidence score [Hyttinen et al., 2014; Magliacane et al., 2016; Hyttinen et al., 2013]. SAT-based approaches handle conflicting independence tests more robustly than traditional constraint-based methods, and support latent variables and useful feedback cycles in causal graphs while maintaining soundness under perfect inputs [Hyttinen et al., 2013; Magliacane et al., 2016]. However, SAT-based approaches remain computationally expensive and intractable beyond small domains.

Score-based methods to CSD evaluate possible DAGs with penalized forms of likelihood. These approaches solve CSD efficiently by performing either greedy hill-climbing search [Tsamardinos et al., 2006; Chickering, 2003; De Campos and Ji, 2011] or constrained optimization using integer linear programs (ILP) [Jaakkola et al., 2010; Cussens, 2011; Yuan et al., 2013; Burlett and Cussens, 2017]. ILP methods can perform exact inference [Bartlett and Cussens, 2017] but require constraints on the number of parents per variable, which are unknown or hard to justify in less understood biological domains.

3 Joint Probabilistic Causal Structure Discovery

The input to causal structure discovery (CSD) is a set \( \mathbf{V} = \{ V_1, \ldots, V_n \} \) of \( n \) variables and \( m \) independent observations of \( \mathbf{V} \). Here, we assume that the observations are drawn without selection bias or hidden confounders, as in PC and most score-based methods. The problem of CSD is to infer a directed acyclic graph (DAG) \( \mathcal{G}^* = (\mathbf{V}, \mathbf{E}) \) such that each edge \( E_{ij} \in \mathbf{E} \) corresponds to \( V_i \) being a direct cause of \( V_j \). If \( V_i \) is a direct cause of \( V_j \), manipulating the value of \( V_i \) changes the marginal distribution of \( V_j \). If \( V_i \) is an ancestor of \( V_k \), there exists a directed path \( p \), denoted by sequence of edges \( V_i \rightarrow \cdots \rightarrow V_k \), from \( V_i \) to \( V_k \). Ancestral structure is encoded by DAG \( \mathcal{G}_A^* \), where edges represent ancestral relations and correspond to the transitive closure of the causal graph \( \mathcal{G}^* \). Typically, CSD methods output an equivalence class of \( \mathcal{G}^* \) and \( \mathcal{G}_A^* \) that correspond to the optimal distribution.

The joint probabilistic CSD problem is to infer causal graph \( \mathcal{G}^* \) together with the ancestral graph \( \mathcal{G}_A^* \). The problem requires defining a suitable joint meta-distribution \( \mathcal{P} \) over the space of possible structures \( \mathcal{G} \) and \( \mathcal{G}_A \). The inputs to \( \mathcal{P} \) are random variables that capture structural and independence attributes of \( \mathcal{G} \) and \( \mathcal{G}_A \). To avoid confusion with the random variables in our probabilistic model, henceforth, we refer to the domain variables \( \mathbf{V} \in \mathbf{V} \) as vertices.

\( C \) and \( A \) are the set of variables \( C_{ij} \) and \( A_{ij} \) for all \( V_i, V_j \) that denote the absence or presence of an ancestral or causal edge, respectively. The goal of inference is to find assignments for these variables. \( U \) is the set of observed variables \( U_{ij} \) associated with an undirected edge, or adjacency, from \( V_i \) to \( V_j \) for all \( V_i, V_j \). \( U \) corresponds to the skeleton graph used in constraint-based methods. The set \( \mathbf{M} \) of \( M_{ij} \) variables denotes marginal association between \( V_i \) and \( V_j \) where each \( M_{ij} \) is obtained by performing a statistical test of independence \( V_i \perp \hspace{-0.15cm} \perp V_j \). Similarly, \( S_{ij} = \{ S^1_{ij}, \ldots, S^m_{ij} \} \) denotes the set of variables that measure conditional association between \( V_i \) and \( V_j \) when conditioned on a non-empty subset of vertices \( \mathbf{Z}_k \subset \mathbf{V}_i \setminus \{ V_i, V_j \} \). Each set \( \mathbf{Z}_k \) has size between 1 and \( |\mathbf{V}| - 2 \). Each \( S^m_{ij} \) corresponds to a statistical test for \( V_i \perp \hspace{-0.15cm} \perp V_j | \mathbf{Z}_m \).

Finally, we optionally observe \( \mathbf{L} = \{ L_{ij}, \ldots, L_{st} \} \), local evidence that captures domain knowledge or side information about causal, ancestral or adjacency relations.

To solve the joint probabilistic CSD problem, the meta-distribution \( \mathcal{P}(C, A, U, S, M, L) \) is first fully defined. Then, we perform maximum a posteriori (MAP) inference over \( \mathcal{P} \) to find an optimal joint assignment to variables \( C \) and \( A \).

4 CAUSPSL Approach

Defining meta-distribution \( \mathcal{P} \) that relates \( \mathcal{G} \) and \( \mathcal{G}_A \) requires a flexible modeling framework. To efficiently solve the joint probabilistic CSD problem, \( \mathcal{P} \) must admit tractable inference. Our approach uses probabilistic soft logic (PSL) [Bach et al., 2017], which offers both desired features. We provide a brief overview of PSL and direct readers to [Bach et al., 2017] for a full description.

4.1 Probabilistic Soft Logic

PSL is a probabilistic programming framework where variables are represented as logical atoms and dependencies between them are encoded via rules in first-order logic. Logical atoms in PSL take continuous values and rule satisfaction is computed using the Lukasiewicz relaxation of Boolean logic. This relaxation into continuous space allows MAP inference to be formulated as a convex optimization problem that can be solved in polynomial time.

Given continuous evidence variables \( \mathbf{X} \) and unobserved variables \( \mathbf{Y} \), PSL defines the following Markov network, called a hinge-loss Markov random field (HL-MRF), over continuous assignments to \( \mathbf{Y} \):

\[
\mathcal{P}(\mathbf{Y} = y | \mathbf{X} = x) = \frac{1}{\mathcal{Z}} \exp \left( - \sum_{r=1}^{M} w_r \phi_r(y, x) \right)
\]

where \( \mathcal{Z} \) is a normalization constant and \( \phi_r(y, x) = \max \{ l_r(y, x), 0 \} \) is an efficient-to-optimize hinge-loss feature function that scores configurations of assignments to \( \mathbf{X} \) and \( \mathbf{Y} \) as a linear function \( l_r(y, x) \) of the variable assignments.
To obtain an HL-MRF, we substitute variables appearing in the first-order logic rules with constants from observations, producing \( M \) ground rules. We observe truth values \( \in [0, 1] \) for a subset of the ground atoms, \( X \) and infer values for the remaining unobserved ground atoms, \( Y \). The ground rules and their corresponding weights map to \( \phi_r \) and \( w_r \). To derive \( \phi_r(y, x) \), the Lukasiewicz relaxation is applied to each ground rule to derive a hinge penalty function over \( y \) for violating the rule. Thus, MAP inference minimizes the weighted rule penalties to find the minimally violating joint assignment for all the unobserved variables. PSL uses the consensus-based ADMM algorithm to perform exact MAP inference.

### 4.2 CAUSPSL Model

CAUSPSL represents statistical tests, causal and ancestral relations as predicates to form orientation constraints in a HL-MRF using the rules shown in Table 1.

#### Predicates

The targets of joint probabilistic inference, \( C_{ij} \) and \( A_{ij} \), are represented with predicates \( \text{CAUSES}(A, B) \) and \( \text{ANC}(A, B) \). We represent undirected edges \( U_{AB} \) with \( \text{Adj}(A, B) \).

We introduce \( \text{ASSOC}(A, B) \) and \( \text{INDEP}(A, B) \) to capture marginal association and independence, corresponding to \( M_{AB} \). \( \text{CONDASSOC}(A, B, S) \) and \( \text{CONDINDEP}(A, B, S) \) denote conditional association and independence, where \( S \) will be substituted with all possible conditioning sets \( Z_m \). These logical atoms correspond to the \( S_{AB} \). To obtain substitutions for these predicates, we enumerate pairwise marginal and conditional tests with all possible conditioning sets up to a maximum size. We threshold \( p \) values from statistical tests to determine whether independence statements are characterized as \( \text{ASSOC}, \text{CONDASSOC} \) or \( \text{INDEP}, \text{CONDINDEP} \). We use \( 1 - p \) as truth values for \( \text{CONDASSOC}, \text{ASSOC} \) and \( \text{CONDINDEP}, \text{INDEP} \). Since adjacencies imply dependence between variables, we obtain \( \text{Adj}(A, B) \) by retaining \( \text{ASSOC}(A, B) \) observations that are never conditionally independent. Finally, because orientation constraints require membership checks in conditioning sets \( S \), we use auxiliary predicate \( \text{INSET}(C, S) \) to indicate that vertex \( C \) is in conditioning set \( S \).

**Local** \( \lambda(A, B) \) predicates denote evidence from source \( \lambda \) for causal, ancestral or undirected edge between vertices \( A \) and \( B \) and correspond to variables \( L \). Obtaining local evidence is domain-specific, and in our experimental evaluation, we show applications of both intervention-based and other side information.

#### Soft Constraints

Table 1 shows the rules used in CAUSPSL. The causal orientation rules (C1-C6) follow from the three sound and complete PC rules [Spirtes and Glymour, 1991] and the ancestral orientation rules (A1-A8) are derived from constraints used in the SAT-based ancestral causal inference (ACI) algorithm [Magliacane et al., 2016]. The basic joint rules (J1-J5) connect ancestral and causal edge predictions through fundamental relationships between the structures introduced in Section 3. These multiple types of well-studied constraints propagate consistency across predictions for CAUSPSL.

**Causal Orientation Rules** Rule C1 discourages causal edges between vertices that are not adjacent. Rule C2 penalizes simple cycles between two vertices. The remaining rules ensure that observed independences match those implied by the graph through d-separation. Rules C3 and C4 correspond to the PC rule which orients chain \( V_i \to V_j \to V_k \) as \( V_i \to V_j \leftarrow V_k \) if conditioning on \( V_j \) breaks the independence between \( V_i \) and \( V_k \). Unlike in PC, in CAUSPSL, \( V_j \) appears in multiple conditioning sets. The redundancy recovers information when \( V_j \) is incorrectly missing from a separating set. Rule C5 captures the PC rule that orients path \( V_i \to V_j \to V_k \) as \( V_i \to V_j \leftarrow V_k \) when \( V_i \to V_j \) is probable and \( V_j \) induces conditional independence between \( V_i \) and \( V_k \). Rule C6 maps to the final PC rule, and if \( V_i \to V_j \to V_k \) and \( V_j \to V_k \), orients \( V_i \to V_k \) to avoid a cycle. PC applies these rules iteratively to fix edges whereas in CAUSPSL, the rules induce dependencies between causal edges to encourage parsimonious joint inferences.
Basic Joint Rules  
Rule J1 encodes that causal edges are also ancestral by definition and rule J2 is its contrapositive. Rule J3 encodes transitivity of ancestral edges, encouraging consistency across predictions. Rule J4 infers causal edges between probable ancestral edges that are adjacent. These four rules exactly encode the relationship between causal and ancestral graphs, and suffice to recover structure under perfect inputs. However, in noisy settings, we gain robustness by including additional joint constraints such as rule J5 and ancestral rules below to recover consistent explanations from conflicting inputs. Rule J5 orients chain ancestral rules below to recover consistent explanations from diverging and linear paths.

Ancestral Orientation Rules  
Ancestral rules A1 and A2 are analogous to their causal orientation counterparts. Rules A3 to A7 follow from lemmas relating minimal conditional (in)dependence to the existence or absence of ancestral edges [Magliacane et al., 2016; Claassen and Heskes, 2011]. Minimal conditional independence is defined as \( (X \perp \perp Y | Z, W) \) and corresponds to ancestral edge existence between \( Z \) and \( X \) or \( Y \). Similarly, minimal conditional dependence is \( \neg(X \perp \perp Y | Z, W) \) and denotes ancestral edge absence between \( Z \), and \( X \) and \( Y \). For compactness, we encode minimal conditional (in)dependence by only comparing marginal associations to conditional tests of set size one. We model ancestral edge existence with three rules, A5 to A7, for each path orientation case: 1) \( \Leftarrow X \to Z \) where \( Z \) is diverging, 2) \( \Leftarrow Z \to X \) where \( Z \) is along linear path from \( X \) to \( Y \), and 3) \( \to Z \to X \) where \( Z \) is along a linear path in the opposite direction. Rule A8 translates the first novel ancestral rule introduced in ACI [Magliacane et al., 2016]. Rules A5 to A8 introduce dependencies across ancestral edge predictions, requiring collective inferences.

5 Experimental Evaluation

Our evaluation demonstrates three advantages of our method: the flexibility of combining multiple structural constraints, scalability for large causal networks, and robustness to noise.\(^1\)

We evaluate our model on standard synthetic data [Hytinnen et al., 2014; 2013; Magliacane et al., 2016] and two real-world biological datasets. We compare against PC [Spirtes and Glymour, 1991], the canonical constraint-based CSD method and Max-Min Hill Climbing (MMHC), a score-based hybrid approach that uses the max-min parents children (MMPC) graph pruning algorithm and has achieved state-of-the-art performance in multiple BN structure learning domains [Tsamardinos et al., 2006]. We also include comparisons against a bootstrapped variant of PC commonly used to improve robustness [Ramsey, 2010; Magliacane et al., 2016]. In our experiments, scalability prevents us from comparing against the SAT-based CSD approach [Hytinnen et al., 2014], which becomes prohibitively expensive for domains larger than eight variables.

\(^1\)Code and data at: bitbucket.org/linqs/causpsl.

5.1 Datasets

We validate our approach using three datasets: (1) synthetic linear acyclic models with Gaussian noise; (2) simulated gene expression from the DREAM challenge [Marbach et al., 2010; Prill et al., 2010]; (3) perturbation experiments on protein-signaling pathways [Sachs et al., 2005].

Synthetic data

To generate synthetic observations, as in previous work [Hytinnen et al., 2014; Magliacane et al., 2016; Hyttinen et al., 2013], we randomly generate 100 ground truth DAGs over 15 variables with edge probability of 0.2 using the pcalg package. We sample 500 observations from each using a linear Gaussian model. CSD methods typically evaluate on this low-noise synthetic setting which serves as a contrast to the more realistic noisy settings described below.

DREAM4 Challenge

Our second dataset from the DREAM4 challenge consists of a gold-standard yeast transcriptional regulatory network and simulated gene expression measurements [Marbach et al., 2010; Prill et al., 2010]. For cross validation, we sample 10 subnetworks of sizes 20 and 30, denoted DREAM20 and DREAM30, with low Jaccard overlap. The real-valued gene expression measurements are simulated from differential equation models of the system at 210 time points. We perform independence tests on the measurements which yield numerous spurious correlations. Additionally, we include domain knowledge of undirected protein-protein interaction (PPI) edges modeled by \( \text{ANC}(A, B) \land \text{LOCAL}_{\text{PPS}}(A, B) \rightarrow \text{CAUSES}(A, B) \).

Protein Signaling Pathway in Human T-Cells

Our third dataset comes from a protein-signaling pathway in human T-cells with flow cytometry measurements [Sachs et al., 2005]. The discovered protein signaling network has been biologically validated and used extensively as a benchmark for evaluating CSD algorithms [Triantafillou and Tsamardinos, 2015; Magliacane et al., 2016; Mooij and Heskes, 2013; Eaton and Murphy, 2007; Peters et al., 2016]. The variables are abundance levels of 11 molecules, measured across eight experimental conditions with 700 to 900 observations each. The first condition activates the pathway and is considered by previous work as the steady-state observed data. The remaining conditions are interventions on seven out of 11 proteins. Following prior work, we consider statistically significant (\( \alpha = 0.05 \)) post-interventional changes as evidence of an ancestral relation between the intervention target and effected protein [Magliacane et al., 2016; Sachs et al., 2005]. We model this intervention-based local evidence as \( \text{LOCAL}_{\text{INTERVENTION}}(A, B) \rightarrow \text{ANCESTOR}(A, B) \).

5.2 Experimental Setup

To evaluate the result quality across methods and robustness to noise, we compute \( F_1 \) scores of predicted causal edges against the ground truth edges from each dataset. To calculate \( F_1 \) in DREAM and synthetic settings, rounding thresholds on the continuous outputs of \( \text{CAUSPSL} \) and Bootstrapped PC are selected using cross-validation with 10 and 100 folds, respectively. In the Sachs setting where the small network...
size prevents sampling of subnetworks for cross-validation, a standard 0.5 threshold is used. For independence tests in all settings, we use linear and partial correlations with Fisher’s Z transformation for continuous data. We run both PC variants and MMHC with the pcalg and bnlearn R packages, respectively. CAUSPL uses ADMM inference implemented in PSL [Bach et al., 2017]. Without a priori preferences for rules, we set all CAUSPL rule weights to 5.0 except for causal and ancestral orientation rules 2 which are set to 10.0, since they encode strong asymmetry constraints. For both PC variants and CAUSPL, we condition on sets up to size two for DREAM20 and up to size one for DREAM30. The MMPC phase of MMHC performs tests on sets up to size $|V| - 2$. For Bootstrapped PC, we follow the bootstrapping procedure used by [Magliacane et al., 2016] and randomly sample 50% of the observations to include in 100 iteration of PC and average the predictions across multiple runs. In DREAM and synthetic settings, $\alpha$ thresholds on independence tests for all methods are also selected within the cross-validation framework. Baselines use $\alpha$ to prune undirected edges while CAUSPL uses separate $\alpha$ values to categorize association tests and identify ADJ. Since $\alpha$ is typically small, we rescale truth values $p$ for CONDINDEP, INDEP by $\sqrt{p}$ to reduce right-skewness of values. For Sachs, we use $\alpha = 0.05$ for all methods, which has been reported to have the best performance in prior work. We rescale $p$-values of the post-interventional changes with the sigmoid function to prevent overconfident local evidence.

### 5.3 Cross-validation Study of Modeling Components

Our first evaluation investigates how each type of constraint in CAUSPL bolsters performance. CAUSPL has three critical modeling components that contribute in differing degrees to improvements in CSD: 1) CAUSPL-PC; 2) CAUSPL-JOINT; and 3) CAUSPL-ANC. Using only the causal edge orientation rules, CAUSPL-PC upgrades PC with multiple independence tests and collective inferences. The CAUSPL-JOINT model combines CAUSPL-PC and basic joint rules for longer-range structural consistency but excludes full ancestral modeling. The CAUSPL-ANC model extends CAUSPL-JOINT with ancestral orientation rules. Finally, we distinguish between CAUSPL-ANC and the complete CAUSPL model, which includes the novel ACI constraint [Magliacane et al., 2016]. To understand the factors affecting result quality, we perform cross-validation across the model variants of CAUSPL and compare against both PC variants and MMHC in the DREAM4 and synthetic settings. Table 2 shows average $F_1$ scores across all methods and datasets.

CAUSPL-PC alone outperforms PC in all settings, with significant gains over both PC variants in two. These improvements suggest that collective inference and multiple statistical tests without pruning alone provide robustness benefits, even over bootstrapping the PC algorithm. CAUSPL-JOINT outperforms CAUSPL-PC in two of three settings, suggesting that modeling even transitivity and short-range dependencies between ancestral and causal structures improves performance. CAUSPL-ANC and CAUSPL further gain over CAUSPL-JOINT in two of three settings. CAUSPL achieves the best performance in DREAM20 with significant gains over MMHC and PC. CAUSPL-ANC outperforms all methods in DREAM30 with gains of up to 50% over both PC variants and 20% over MMHC. Our best performing PSL models significantly outperform multiple baselines using a paired t-test on DREAM, showing the benefit of more sophisticated ancestral-causal constraints under noisy experimental conditions, where spurious correlations dominate. On straightforward linear Gaussian data, all modeling variants of CAUSPL significantly outperform both PC variants and MMHC with $F_1$ score improvements of up to 17.5%. However, in this synthetic setting, simpler CAUSPL-PC and CAUSPL-JOINT models suffice for good performance. The contrasting result highlights the importance of evaluating CSD methods on more realistic settings.

### 5.4 Comparisons in Real-World Sachs Setting

In the real-world Sachs setting, we compare the $F_1$ scores of causal edge predictions by CAUSPL-ANC and CAUSPL against those of MMHC, the best performing baseline method. Additionally, we compare our ancestral edge predictions to ACI results reported by [Magliacane et al., 2016]. CAUSPL-ANC improves over MMHC from 0.307 to 0.32 $F_1$ while CAUSPL performs as well as MMHC. For ancestral inference, ACI achieves a reported $F_1$ score of 0.38. CAUSPL-ANC gains over ACI with an $F_1$ of 0.43 and CAUSPL also improves over ACI with a score of 0.4.

### 5.5 Scalability

Our second evaluation focuses on the scalability of our approach. PC and MMHC scale by iteratively pruning adjacencies with statistical tests, potentially sacrificing result quality despite permitting larger conditioning set sizes. More flexible SAT-based methods enumerate all statistical tests but cannot scale to large networks. For example, running the SAT approach proposed by [Hyttinen et al., 2014] with nine variables and a conditioning set size of one required over 40 minutes [Magliacane et al., 2016]. In contrast, CAUSPL uses all statistical tests without pruning and requires less than a

---

<table>
<thead>
<tr>
<th>Dataset</th>
<th>PC</th>
<th>MMHC</th>
<th>Bootstrapped PC</th>
<th>CAUSPL-PC</th>
<th>CAUSPL-JOINT</th>
<th>CAUSPL-ANC</th>
<th>CAUSPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synth</td>
<td>0.74 ± 0.09</td>
<td>0.76 ± 0.12</td>
<td>0.72 ± 0.11</td>
<td>0.87 ± 0.06</td>
<td>0.87 ± 0.06</td>
<td>0.86 ± 0.06</td>
<td>0.87 ± 0.06</td>
</tr>
<tr>
<td>DREAM20</td>
<td>0.15 ± 0.04</td>
<td>0.17 ± 0.05</td>
<td>0.18 ± 0.05</td>
<td>0.17 ± 0.05</td>
<td>0.18 ± 0.05</td>
<td>0.19 ± 0.05</td>
<td>0.20 ± 0.05</td>
</tr>
<tr>
<td>DREAM30</td>
<td>0.16 ± 0.03</td>
<td>0.2 ± 0.05</td>
<td>0.16 ± 0.04</td>
<td>0.22 ± 0.03</td>
<td>0.23 ± 0.03</td>
<td><strong>0.24 ± 0.03</strong></td>
<td>0.22 ± 0.03</td>
</tr>
</tbody>
</table>

Table 2: Average $F_1$ scores of methods across datasets. We show how each CAUSPL component contributes to performance.
second for 10 variables, overcoming the inference scalability bottleneck. To evaluate running times, we generate synthetic linear Gaussian networks and sample DREAM4 subnetworks of increasing size. Our method computes all possible statistical tests up to conditioning set size denoted by $C$ and the baseline methods prune conditioning sets through independence. In Table 3, we present running times for all methods, splitting up our approach into conditional independence testing (CI) and inference (Inf). We show that CausPSL can efficiently infer causal graphs while using more information than competing methods.

The running time depends on the network size $n$ and the maximum conditioning set size $C$. The results indicate that the dominant factor in the running time of our method is enumerating all statistical tests rather than inference. For the largest networks ($n = 50, C = 1$), computing statistical tests requires approximately a minute, while inference only requires 7 to 10 seconds. Larger conditioning sets impact running time, requiring up to 10 minutes when $n = 30, C = 2$. However, Table 2 shows that by enumerating statistical tests, CausPSL outperforms pruning-based methods with only $C = 1$. SAT-based methods also enjoy this benefit [Magliacane et al., 2016] but require expensive inference. In contrast, CausPSL completes inference within 10 seconds for 30- and 50-variable networks when $C = 1$. In further study, CausPSL completed inference for a DREAM4 network with 100 variables in 27 minutes, scaling to an order of greater magnitude than SAT-based methods. In future work, statistical tests can be parallelized to admit larger $C$.

### 5.6 Robustness to Noisy Evidence

In our final evaluation, we validate the robustness of CausPSL to imperfect evidence. CausPSL incorporates real-valued noisy signals within joint inference, exploiting global structural constraints to smooth local errors. In contrast, MMHC and PC must discretize noisy evidence and incorporate domain knowledge as fixed edges or non-edges.

To evaluate the robustness of CSD methods on DREAM30 subnetworks, we simulate noise with a new local ancestral signal drawn by fixing a Bernoulli error rate and sampling real-valued evidence from its conjugate, a $\beta$ distribution. We set a Bernoulli error rate of $1 - p$. For each pair of vertices, with probability $p$, true ancestral edges are sampled from $\beta(8, 2)$, and true non-edges are sampled from $\beta(2, 8)$ which are peaked at high-confidence and accurate soft truth values. With probability $1 - p$, incorrect values are sampled from $\beta(2, 5)$ and $\beta(5, 2)$ for edges and non-edges, respectively. For CausPSL, we incorporate this new signal using the local ancestral evidence rule shown in the Sachs setting. For MMHC and PC, synthetic values from this signal of $< 0.5$ are treated as fixed causal non-edges, representing the hard version of joint rule J3 in Table 1. Synthetic values $\geq 0.5$ are intersected with PPI edges to obtain fixed causal edges, simulating the discretized version of the PPI rule given in the DREAM setting.

In Fig. 1, we compare average $F_1$ scores across all modified methods as the Bernoulli error rate of the synthetic signal increases from 0.0 to 0.9. CausPSL remains robust as the error increases beyond 0.3 while PC and MMHC steadily degrade in performance. When the signal is near-perfect with error $\leq 0.2$, the baselines receive select correct causal edges while CausPSL fuses the signal with imperfect statistical tests. However, analysis of intervention-based evidence in the Sachs setting shows that real-world local signals are in the $\geq 0.5$ noise regime, where CausPSL excels over compared methods.

### 6 Discussion and Conclusion

We propose a probabilistic model for the CSD problem that achieves scalability despite using multiple independence tests and global structural constraints. Our method is flexible, fusing noisy ancestral and causal signals with side information from PPI networks and interventions. Our experimental highlights include: 1) scaling up to networks with hundreds of variables; 2) achieving significant performance gains over constraint- and score-based baselines despite many spurious correlations; and 3) showing robustness to increasingly noisy local signals. In future work, we will extend our approach to support latent variables and perform approximate marginal inference to score possible causal and ancestral edges.

Table 3: Running times in seconds for obtaining conditional independence tests (CI) and inference (Inf). CausPSL scales to large networks using multiple tests with no pruning.

<table>
<thead>
<tr>
<th>D Size</th>
<th>PC</th>
<th>MMHC</th>
<th>PSL: $C=1$</th>
<th>PSL: $C=2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CI</td>
<td>Inf</td>
<td>CI</td>
<td>Inf</td>
</tr>
<tr>
<td>Synth</td>
<td>10</td>
<td>0.02</td>
<td>0.07</td>
<td>0.19</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.06</td>
<td>0.03</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0.19</td>
<td>0.15</td>
<td>4.94</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.44</td>
<td>0.48</td>
<td>65.4</td>
</tr>
<tr>
<td>DREAM4</td>
<td>10</td>
<td>0.03</td>
<td>0.06</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.08</td>
<td>0.06</td>
<td>0.73</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0.22</td>
<td>0.15</td>
<td>3.76</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.41</td>
<td>0.49</td>
<td>57.1</td>
</tr>
</tbody>
</table>

Figure 1: Average $F_1$ score vs. synthetic evidence noise rate on DREAM4 ($n = 30, C = 1$). CausPSL remains robust as noise rate increases.
Acknowledgements

This work is sponsored by Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA), including contract FA8650-17-C-7715, and supported by NSF grants CCF-1740850 and NSF IIS-1703331.

References


