INTEGRATING OVERLAPPING STATIC AND TIME-SERIES CAUSAL GRAPHS VIA CONSTRAINT-SOLVING

Anonymous authors
Paper under double-blind review

ABSTRACT

In many causal learning problems, variables of interest are not always measured in the same observations, but are instead distributed across multiple datasets with overlapping variables. Previous algorithms used independence tests to directly determine the minimal equivalence class of DAGs consistent with all input graphs, but these methods faced significant computational limits. In this paper, we formulate this problem as a more computationally efficient answer set programming (ASP) problem called SCION (Solving Constraints for Integration of Overlapping Networks), which can be solved with the ASP system *clingo*. We then extend SCION to a novel method, SCION-TS, for solving the overlapping variables problem in the timeseries setting. We test both algorithms across multiple simulations to demonstrate their computational viability, and successfully apply both algorithms to input data, using p-value pooling to improve input graph estimation.

1 INTRODUCTION

Many inference problems require the use of data from different sources. Ideally, these data would be merged in a single unified dataset, but that might not be possible for financial, legal, ethical, or other reasons. For example, we might want a unified model using datasets from financial and healthcare organizations to find relationships between health and financial outcomes. In practice, these datasets likely cannot be integrated for privacy reasons (or because they include different samples), though some variables (e.g., age) are presumably measured in both. This problem arises for both static and time-series data, though the latter is more complicated due to more complex induced time-lagged relationships between observed variables in each subgraph. In this work, we develop methods for this problem, learning causal relationships between variables not co-measured in any dataset.

Formally, we seek the complete set of graphs $\mathbb{H}_i \in \mathcal{H}$ over all variables that are consistent with a set of input graphs $\mathbb{G}_i \in \mathcal{G}$, each learned locally from a source dataset.[1] The first algorithm for this type of problem, Integration of Overlapping Networks (ION) Tillman et al. (2008), constructively used input graph independence relations to iterate through sets of changes that preserved those relations. This approach was computationally expensive, limiting testing to 4- and 6-node global graphs.

In this paper, we present **SCION: Solving Constraints for Integration of Overlapping Networks**: a more efficient answer-set programming (ASP) formulation that is complete and sound. We then extend SCION to **SCION-TS**, the first-ever method that integrates *time series* graphs with inferred longer lags. We use SCION-TS to produce the first constructive characterization of equivalence classes for time series causal graphs with latent variables. Section 2 describes related work. Section 3 presents the ASP formulation of SCION for static data, evaluates it on synthetic input, and applies it to real-world data from the European Social Survey. Section 4 presents SCION-TS for time series data, and evaluates it on simulated data. Section 5 considers limitations and extensions.

2 RELATED WORK

Most structure learning methods (causal or otherwise) take a single dataset as input, so one approach when provided multiple overlapping datasets is to somehow integrate them together. Statistical

[1] We assume: for any $\mathbb{G}_j, \mathbb{G}_k$, there is a sequence of input graphs from $\mathbb{G}_j$ to $\mathbb{G}_k$ such that each consecutive pair of graphs in the sequence have non-empty intersection of their variable sets.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

matching approaches match observations across datasets using distance in the co-measured features (Budd & Radner, 1969; Okner, 1972). These matches enable imputations of variable values, or other statistical information about non-co-measured variables (Leulescu & Agafitei, 2013). Traditional statistical matching approaches are only provably reliable under a strong independence assumption: if $\mathbf{V}_1$, $\mathbf{V}_2$, and $\mathbf{V}_c$ are the variables in only datasets 1, 2, and their intersection respectively, then we must have $\mathbf{V}_1 \perp\!\!\!\perp \mathbf{V}_2 \mid \mathbf{V}_c$. This assumption is both rarely true in practice, and untestable given only the input datasets (Sims, 1972; Rodgers, 1984). Methods without this assumption require additional data (Paass, 1986; Singh et al., 1993), or informative proxies (Zhang, 2015).

Federated learning (FL) methods seek to efficiently learn a single model, not all consistent global models, from multiple data sources. They typically assume one could (in theory) construct a unified dataset, but they seek to minimize data exchanges (Kairouz et al., 2021). In "horizontal" FL problems, each data source contains a partition of observations over a shared feature space; in "vertical" FL, data sources contain different features about shared observations (Yang et al., 2019). Federated "transfer" learning approaches aim to learn from data sources with different sets of features and observations, but typically with some overlap in observations (Liu et al., 2020; Sharma et al., 2019).

Our paper is closest to Tillman et al. (2008), whose ION algorithm is an asymptotically correct algorithm that outputs the equivalence class of directed acylic graphs (DAGs) consistent with an input set of overlapping partial ancestral graphs (PAGs). ION constructively builds, in a branch-and-prune manner, the full set of global DAGs that preserve all d-separation and d-connection relations in the input PAGs. ION is complete and sound, but requires superexponentially many operations due to its constructive structure and the problem being NP-complete. As a result, Tillman et al. (2008) only ran ION on highly overlapping 4- and 6-node ground-truth DAGs.

The subsequent Integration of Overlapping Datasets (IOD) algorithm (Tillman & Spirtes, 2011) takes datasets as input rather than PAGs, and derives independence information from p-value pooling over multiple datasets (see Section 3.2) rather than input PAGs. IOD outperforms ION in memory, precision and recall, largely because IOD smoothly resolves (statistical) inconsistencies between input datasets. Huang et al. (2020) showed that the global DAG is identifiable from data, if we have a linear data-generating process with non-Gaussian noise and if every pair of datasets has at least two variables of overlap (or only one variable if noise distributions are fixed). They also provide estimation approaches based on expectation-maximization and on adversarial learning.

Boolean satisfiability (SAT) solvers have also been applied to versions of this problem. Triantafillou et al. (2010) used a SAT solver to find a single graph with all pairwise causal relationships consistent with input graphs. Hyttinen et al. (2013) used a SAT formulation of d-separation to discover cyclic causal models from overlapping input graphs. The COmbINE algorithm (Triantafillou & Tsamardinos, 2015) used a SAT-based approach for datasets derived from randomized controlled trials, weighting constraints by p-values. Our approach uses answer set programming (ASP), a declarative problem-solving framework in which logical rules describe solution conditions for the problem (Marek & Truszczyński, 1999; Gelfond & Lifschitz, 1988). ASP leverages optimizations for the boolean SAT problem (Gebser et al., 2007) in problem formulations with high expressiveness (Eiter et al., 2009; Brewka et al., 2011). ASP-based approaches encode background causal knowledge with conditions in the solution program, and have been used for other causal learning problems (Sonntag et al., 2015; Rantanen et al., 2020; Abavisani et al., 2023; Solovyeva et al., 2023).

Finally, for time series data, Gomez et al. (2025) use a score-based approach for the overlapping datasets problem, iteratively combining learned (local) causal models subject to overall consistency, until all models have been combined. However, this method returns only a single output model.

3 SCION FOR STATIC NETWORKS

We first consider the (simpler) case of static data. SCION takes as input a set of overlapping PAGs $\mathbb{G}_i \in \mathcal{G}$, where there is a sequence of graphs from $\mathbb{G}_i$ to $\mathbb{G}_j$ for all i, j such that each consecutive pair of graphs has a non-empty intersection (cf. footnote 1). SCION outputs a complete set of DAGs $\mathbb{H}_i \in \mathcal{H}$, each over the union of all nodes in input PAGs, that each preserve all d-separation/connection relations and edge directions in every input PAG $\mathbb{G}_i$. As an example, if $\mathbb{G}_1 = X \rightarrow Y \rightarrow Z$ and $\mathbb{G}_2 = X \rightarrow W \rightarrow Z$, then exactly two global graphs (over $\{W, X, Y, Z\}$) preserve d-separation/connection and edge directions in the inputs:

Listing 1: *clingo* problem specification for SCION problem.

```
1  {true_edge(X,Y)} :- node(X), node(Y).
2
3  :- true_edge(X, Y), X = Y.
4  :- true_edge(X, Y), nedge(X, Y, T), varin(T, X), varin(T, Y).
5
6  :- true_edge(X, Y), path(Y, X).
7  path(Y, X) :- true_edge(Y,X).
8  path(Y, X) :- true_edge(Y, Z), path(Z, X).
9
10 directed(X, Y, T) :- true_edge(X, Y), varin(T, Y).
11 directed(X, Y, T) :- true_edge(X, Z), directed(Z, Y, T), not varin(T,Z).
12
13 causalconn(X, Y, T) :- directed(X, Y, T).
14 causalconn(X, Y, T) :- directed(Z, X, T), directed(Z, Y, T), not varin(T, Z).
15
16 undirected_causalconn(X,Y,T) :- directed(X, Y, T).
17 undirected_causalconn(X,Y,T) :- directed(Y, X, T).
18 undirected_causalconn(X,Y,T) :- directed(Z, X, T), directed(Z, Y, T), not varin(T, Z).
19
20 bidirected(X, Y, T) :- causalconn(X, Y, T), not directed(X, Y, T), not directed(Y,X,T).
21
22 :- nedge(X, Y, T), undirected_causalconn(X, Y, T), varin(T,X), varin(T, Y).
23 :- input_edge(X, Y, T), not directed(X, Y, T), varin(T, X), varin(T, Y).
24
25 #show true_edge/2.
```

$\mathbb{H} = \{X \to Y \to W \to Z, X \to W \to Y \to Z\}$. We thus learn that Y and W are adjacent (with unknown orientation), even though Y and W were never jointly measured.

We first present an ASP formulation of what we call the "SCION problem," implemented in *clingo* (Gebser et al., 2019) based on the solver *clasp* (Gebser et al., 2007). This formulation specifies the space of possible output graphs (as possible facts), along with rules that must hold in any valid solution. *clingo* then provably outputs all possible graphs that follow these facts and rules (see Listing 1 for the *clingo* problem specification). Input PAGs are specified using three predicates: (1) `input_edge(X,Y,T).` denotes $X \to Y$ in input PAG T; (2) `bidirected(X,Y,T).` denotes $X \leftrightarrow Y$ in T; and (3) `nedge(X,Y,T).` denotes known absence of an edge between X and Y in T. `varin(T, X).` denotes nodes in T, and the number of input PAGs is also provided.

Line 1 defines any set of edges between nodes as a potential solution. Lines 3-5 specify general constraints: (3) no self-loops; (4) edges absent in an input PAG cannot appear in a solution (since edge absence indicates a persistent conditional independence); and (5) all valid solutions are acyclic. Lines 7/8 and 10/11 recursively define directed paths, and directed edges relative to an input graph T, respectively. Lines 13 and 14 define a causal connection between nodes X and Y in T as either $X \to Y$ in T, or an unobserved-in-T common cause of X, Y. Lines 16 through 18 define an undirected causal connection as $X \to Y$ in T, $Y \to X$, or an unobserved-in-T common cause of X, Y. These two terms are used to represent circle edges $X \circ\!\!\to Y$ and $X \circ\!\!-\!\!\circ Y$ in input PAGs. Line 20 specifies that a bidirected edge in input graph T means a causal connection due to an unobserved-in-T common cause. Lines 22 and 23 provide closure constraints on the inputs: nonexistence of an edge between nodes in T implies no causal connection, and a directed edge between nodes in T implies a directed path, respectively. Line 25 provides the output.

To show that the SCION ASP formulation leads to the correct output equivalence class, we prove in Appendix A that it is both complete and sound with respect to causal relationships in input PAGs.

Theorem 1. *Soundness: If nodes X and Y are d-separated (d-connected) given nodes $\mathbf{Z}$ in some $\mathbb{G}_i \in \mathcal{G}$, then X and Y are d-separated (d-connected) given $\mathbf{Z}$ in every output $\mathbb{H}_i \in \mathcal{H}$.*

Theorem 2. *Completeness: Let $\mathbb{H}_i$ be a partial ancestral graph over variables $\mathbb{V}$ such that for every $\{(X, Y)\} \subseteq \mathbb{V}$, if X and Y are d-separated (d-connected) given $\mathbf{Z} \subseteq \mathbb{V}/\{X, Y\}$ in some $\mathbb{G}_i \in \mathcal{G}$, then X and Y are d-separated (d-connected) given $\mathbf{Z}$ in $\mathbb{H}_i$. Then, $\mathbb{H}_i$ is in $\mathcal{H}$.*

3.1 Simulations with Input PAGs

We tested SCION on graphs of varying sizes, densities, and overlap between subgraphs. We randomly generated "ground truth" DAGs using: (i) total number of nodes $\mathcal{N}$; (ii) DAG density p_d; (iii)

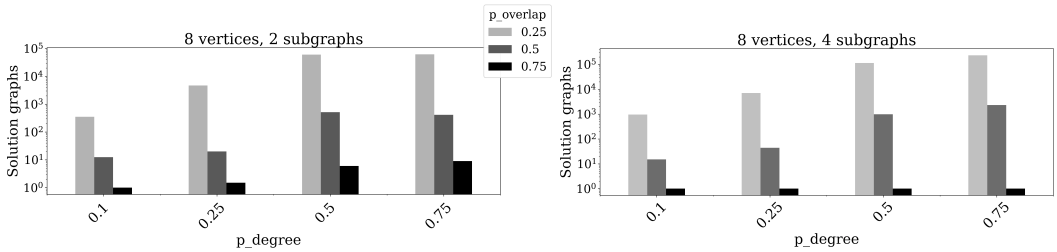


Figure 1: Median number of DAGs in the solution set. Note the logarithmic scale on the y-axis.

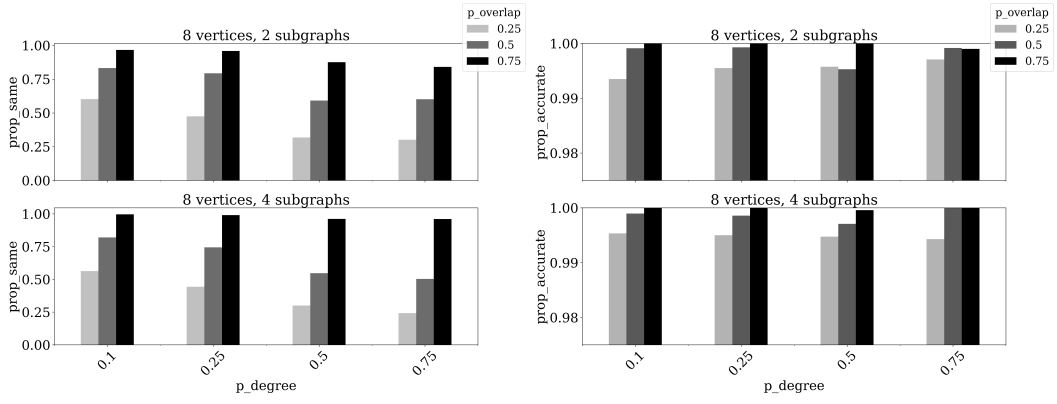


Figure 2: Left: Mean $prop_same$ (90% threshold); Right: Mean $prop_accurate$ (90% threshold).

s input PAGs; and (iv) PAG overlap p_o. Each node is randomly connected to $k \sim \text{Bin}(\mathcal{N}-1, p_d)$ other nodes. Uniform random edges yield significantly harder SCION problems than in many real-world settings, and enable us to understand the empirical complexity of SCION. If the DAG is not (weakly) connected, then we add random edges to connect it. We generate input variable sets by dividing $\mathcal{N}$ into s disjoint sets, and for each, adding p_o random nodes from the others.

These simulations aimed to both demonstrate the computational feasibility of SCION, and better understand the informational content of overlapping datasets (compared to a single unified dataset). We analytically generate inputs by marginalizing out non-input set variables, resulting in input PAGs that are provably causally faithful to the ground truth. For example, if $X \to Z \to Y$, then the input PAG for $\{X, Y\}$ is $X \to Y$. Marginalization can similarly yield $X \leftrightarrow Y$ in input PAGs. Input PAGs are converted into sets of statements (using Listing 1 predicates) and passed to the ASP solver.

We simulated 100 DAGs for each of $\mathcal{N} \in \{6, 8, 10, 15, 25\}$, $p_o \in \{0.25, 0.5, 0.75\}$, $p_d \in \{0.1, 0.25, 0.5, 0.75\}$, and $s \in \{2, 4\}$. Higher densities often timed out for $\mathcal{N} \in \{15, 25\}$, so we added $p_d \in \{0.025, 0.05, 0.075\}$ for those $\mathcal{N}$. SCION and SCION-TS problems were run with 4-hour timeouts for *clingo* solver (excluding input generation and output parsing) on 8-thread nodes with 24GB RAM. We report results for completed runs in the 77 settings with $\geq 95\%$ completion.

We focus on two key performance measures. First, $prop_same$ is the proportion of all possible edges or absences that are shared across 75%, 90%, and 100% of the solution set; it measures the similarity of output DAGs. Second, $prop_accurate$ indicates the proportion of those shared edges/absences (in 75%, 90%, or 100% of the solution set) that are found in the ground-truth DAG. Assuming a uniform prior over solution graphs (as we often lack background knowledge about likelihood of solutions), $prop_accurate$ provide an accuracy measure. In the main text, we report both statistics on adjacencies (merging solution edges of opposite directions), as prior work has found adjacency inferences to be more robust and stable than for orientation. However, results for all parameterizations, including results including edge directions, are reported in supplementary materials. We focus in the main text on 8-, 15-, and 25-node DAGs, as their patterns are representative.

As expected, the most important factor for the number of output graphs (Figure 1), and consequently the algorithm runtime, was the amount of overlap between input PAGs. For example, the rightmost

Table 1: 15-node global graphs, 2 input PAGs: $prop_same$ (left) and $prop_accurate$ for 90% (right)

		$p_{overlap}$		
		0.25	**0.5**	**0.75**
p_{degree}	**0.025**	0.802	0.935	0.992
	0.050	0.782	0.933	0.990
	0.075	*	0.921	0.991
	0.100	*	0.912	0.988
	0.250	*	*	0.961
	0.500	*	*	0.900

		$p_{overlap}$		
		0.25	**0.5**	**0.75**
p_{degree}	**0.025**	0.994	0.999	1.000
	0.050	0.997	1.000	1.000
	0.075	*	0.999	1.000
	0.100	*	1.000	1.000
	0.250	*	*	1.000
	0.500	*	*	1.000

Table 2: 25-node global graphs, 2 input PAGs: $prop_same$ (left) and $prop_accurate$ for 90% (right)

		$p_{overlap}$	
		0.50	**0.75**
p_{degree}	**0.025**	0.705	0.921
	0.050	0.685	0.915
	0.075	*	0.928
	0.100	*	0.917

		$p_{overlap}$	
		0.50	**0.75**
p_{degree}	**0.025**	0.995	0.991
	0.050	0.996	1.000
	0.075	*	0.999
	0.100	*	1.000

bars in the left panel of Figure 1 correspond to 61946, 417, and 9 median solution graphs, respectively. Many parameterizations had a median of 1 output graph, but almost all had a very long runtime tail. For example, for $\mathcal{N} = 15$, two input PAGs, $p_d = 0.05$, and $p_o = 0.25$: 50 graphs yielded solutions within 5.16 seconds; 90 within 648 seconds; but one required over 5400 seconds.

Overlap in input PAGs is the largest factor in the proportion of edge adjacencies or absences shared across 90% of the solution set (Figure 2). Less overlap also typically led to lower accuracy in terms of the widely-shared edges in the output set, though this was not the case for every parameterization. The number of input PAGs, s, had little impact compared to the degree of overlap, with similar results for 2 and 4 PAGs. These patterns are replicated across all values of $\mathcal{N}$ (e.g., Tables 1 and 2).

Denser ground-truth DAGs averaged larger solution sets across all $\mathcal{N}$ and s, and as solution sets get larger, there are fewer shared edges or absences (i.e., slight decreases in $prop_same$ for 90% of solutions). For larger DAGs, this decrease did not generally occur until $p_d \geq 0.1$. $prop_accurate$ is quite high for edges or absences in at least 75% and 90% of solution graphs: 98.86% and 99.85%, respectively. (Those in 100% of outputs were always correct since the input PAGs were correct.)

In order to establish the increased computational efficiency of SCION over the original ION, we pass the same graphs to the ION implementation provided in the `Tetrad` package (Scheines et al., 1998) (available at `https://github.com/cmu-phil/tetrad`) running in the same computational environment (with identical nodes, RAM, and four-hour timeouts). SCION is significantly faster across a range of density and overlap settings among 8-node graphs; for example, with 8-node graphs split into two subgraphs with $p_d = 0.1, p_o = 0.25$, only 13 of 100 runs of ION finished within four hours, while all 100 runs of SCION completed in the same time. In high-overlap settings where ION was able to complete at least 95% of simulations within the timeout, ION was significantly slower than SCION; in 8-node graphs split into two subgraphs with $p_d = 0.1, p_o = 0.75$, the median runtime for the 99 completed ION runs is 0.240 seconds vs. 0.013 seconds for the 100 completed SCION. Full runtime results for ION are available in Appendix B.

3.2 Simulations with Input Data

SCION can, in ideal conditions, extract significant useful information from overlapping input PAGs, but applications typically start with data so we apply SCION to both synthetic data and a real-world social science case. For synthetic data, we used the stochastic block model (Holland et al., 1983) to generate 100 DAGs with clusters of nodes with higher within-cluster density than between-cluster.

We used ground-truth DAGs with $\mathcal{N} = 8$; within-cluster edge density $\rho_w \in \{0.25, 0.5\}$; between-cluster edge density $\rho_b \in \{0.05, 0.1, 0.25\}$; and generated 1000 linear-Gaussian datapoints over all eight nodes (noise $\sigma \in \{0.05, 0.1\}$). We then split the dataset in two, such that each contained the variables from one cluster, plus 1-2 variables from the other.

If the estimated conditional independencies differ between datasets, then the input PAGs can be inconsistent with each other, resulting in an unsatisfiable SCION problem. We thus use the p-value "pooling" approach in Tillman & Spirtes (2011), based on Fisher's 1934 method for combining p-values. This method replaces dataset-specific independence tests for shared variables with merged independence tests: every conditional independence test for shared variables receives a single result that is used in learning all input PAGs. Those input PAGs can require use of the `causalconn` and `undirected_causalconn` atoms in Listing 1 to capture edges with circle endpoints.

We applied FCI (Spirtes et al., 1995) with the Fisher z-test Fisher (1921) from the `causal-learn` package Zheng et al. (2024) and p-value pooling to each pair of input datasets to yield input PAGs for SCION. We converted $X - - Y$ and $X - - \circ Y$ edges into $X \circ\!\!-\!\!\circ Y$ since sample selection bias is not possible in the simulation. Full results, including results using edge direction, are in Appendix C. As an example, consider the simulations with $\rho_w = 0.25$, $\rho_b = 0.25$, $\sigma = 0.05$, and 3 nodes of overlap between subgraphs. Averaged over 100 ground-truth graphs, we find 45.8%, 41.2%, and 40.9% of undirected edge adjacencies and absences were shared in 75, 90, and 100% of the solution set, respectively. These adjacencies and absences were accurate to the ground truth 90.7%, 91.8%, and 91.8% of the time.[2] Overall, for 14 of our 17 parameterizations, we find above 90% accuracy for those edges present in 75% of the solution set.

We also tested the real-world performance and utility of SCION using data from years 2016 and 2018 of the European Social Survey (ESS) (ERIC, 2017; 2019). The ESS, conducted every two years, has a stable core set of questions, along with topical year-specific questions. Rotating modules not asked in the same year are thus not co-measured, but SCION can be used to enumerate possible global graphs. We selected 2 variables from the "welfare attitudes" module of 2016 ESS; 2 from the "justice and fairness" module of 2018 ESS; and 4 variables measured in both survey rounds. There are plausible connections between attitudes about the round-specific topics: for example, someone who is concerned about fairness might want a strong, supportive welfare system.

We applied FCI and p-value pooling to learn input PAGs, and SCION ran for 1.62 seconds to return a solution set with 223,172 graphs (full results in Appendix D). The output graphs significantly overlapped: of 27 possible (global) adjacencies, 15 edge absences were shared across the entire solution set. Interestingly, SCION infers edges between all four pairs of non-co-measured nodes; that is, the ESS data implies that these never co-measured factors are potentially directly causally connected. These possibilities make intuitive sense: for example, dissatisfaction about relative fairness in pay and benefits (measured by *ifrjob*) could plausibly lead to support for a basic income (*basinc*).

4 SCION-TS FOR TIME-SERIES NETWORKS

Data sources often collect time series data, but previous algorithms for the SCION problem focused on static data. We here provide a time-series version of the SCION algorithm, **SCION-TS**, which is the first to find the full equivalence class of global (time-indexed) DAGs given overlapping (time-indexed) PAGs. We assume that Markov order 1 ground truth (i.e., all true edges are $X_{t-1} \to Y_t$), but *measured* lags greater than 1 can result from latent variables not measured in a subgraph. For example, if $X_{t-1} \to W_t$ and $W_{t-1} \to Y_t$, then marginalizing out W induces $X_{t-2} \to Y_t$.

Formally, the inputs are a set of compressed representations of dynamic causal graphs $\mathbb{G}_i \in \mathcal{G}$, where each $X \to Y$ has an associated lag-set $\mathbf{L}$ such that $\forall L_i \in \mathbf{L}, X_{t-L_i} \to Y_t$. Temporal information is encoded in the lagsets, so the compressed graph can contain self-loops (if $X_{t-1} \to X_t$) or edges in both directions between a pair of nodes (if $X_{t-1} \to Y_t$ and $X_t \leftarrow Y_{t-1}$). The inputs can also include bidirected edges representing latent confounders with time lags. For example, if $X \leftarrow W \to Z \to Y$ (all edges lag 1) and only X and Y are observed, then W is a latent confounder but the path from W to Y is one step longer than from W to X. Hence, $X_{t-1} \leftrightarrow Y_t$. Another complication is that the set of observed lags between two variables can be infinite; for example, if $X \to W \to Y$ and only X, Y observed, then if $W_{t-1} \to W_t$, then $X_{t-k} \to Y_t$ for all k.

[2] Statistical noise means that an edge (or absence) found in 100% of output graphs could still be incorrect.

Listing 2: *clingo* specification for SCION-TS problem.

```
1  {true_edge(X,Y)} :- node(X), node(Y).
2
3  :- true_edge(X,Y), not input_edge(X,Y,1,T), varin(T,X), varin(T,Y).
4
5  lag(0..timesteps-1).
6  directed(X,Y,1,T) :- lag(L), true_edge(X, Y), varin(T, Y).
7  directed(X,Y,L,T) :- lag(L), L>1, true_edge(X,Z), directed(Z,Y,L-1,T), not varin(T,Z).
8  common_cause(X,Y,L,T) :- lag(L), directed(Z,X,L1,T), directed(Z,Y,L2,T), not varin(T,Z),
       ↪ varin(T,X), varin(T,Y), L = L2 - L1, not X = Y.
9  common_cause(X,Y,L,T) :- lag(L), directed(Z,X,L1,T), directed(Z,Y,L2,T), not varin(T,Z),
       ↪ varin(T,X), varin(T,Y), L = L2 - L1, X = Y, not L1 = L2.
10
11 directed_causalconn(X, Y, L, T) :- directed(X, Y, L, T).
12 directed_causalconn(X, Y, L, T) :- common_cause(X, Y, L, T).
13
14 :- input_edge(X,Y,L,T), not directed(X,Y,L,T), varin(T,X), varin(T,Y).
15 :- not input_edge(X,Y,L,T), directed(X,Y,L,T), varin(T,X), varin(T,Y).
16
17 input_bidirected(X,Y,L,T) :- input_bidirected(Y,X,L,T), L = 0.
18 :- input_bidirected(X,Y,L,T), not common_cause(X,Y,L,T), varin(T,X), varin(T,Y).
19 :- input_bidirected(X,Y,L,T), not common_cause(X,Y,L,T), X = Y, varin(T,X), varin(T,Y).
20
21 :- not input_bidirected(X,Y,L,T), common_cause(X,Y,L,T), L > 0, varin(T,X), varin(T,Y).
22 both_cc(X,Y,L,T) :- common_cause(X,Y,L,T), common_cause(Y,X,L,T).
23 :- not input_bidirected(X,Y,L,T), both_cc(X,Y,L,T), L=0, varin(T,X), varin(T,Y).
24
25 #show true_edge/2.
```

The SCION-TS inputs are encoded with the same predicates as SCION, though updated to include a time lag set $\mathbf{L}$ for each edge. And rather than represent potentially infinite lags, we bound the maximum lag for each input as "number of nodes not in that input + 1," and provide *clingo* with the maximum represented lag in each graph. The SCION-TS output contains all compressed causal DAGs with Markov order 1 over the total node set that are consistent with the independence and association facts in the input graphs.

Listing 2 provides the *clingo* specification for SCION-TS. Line 1 generates the set of possible solutions. Line 3 encodes a new constraint that provably holds: there is a true $X \to Y$ edge if and only if there is a lag-1 $X \to Y$ in an input where they are co-measured. Line 5 defines the set of possible lags. The remaining lines mirror SCION, but adjusted to account for time lags in the input graphs. Time lags are a mixed blessing: they increase the dimensionality and complexity of the problem, but also provide additional constraints by specifying the lengths of directed paths.

We establish the correctness of SCION-TS in a slightly different fashion, as we show that the above ASP specification is a *direct encoding* of the SCION-TS problem—that is, it exactly characterizes the intended solution set—by establishing that every relationship between nodes in the input graphs is entailed in all valid solutions returned by the SCION-TS algorithm, and that the lack of such a relationship in the input precludes its existence in any output (proof in Appendix A).

Theorem 3. *Listing 2 is a direct encoding of the SCION-TS problem.*

Prior work showed that *clingo* returns a complete solution set given a direct encoding as input if the program considers all possible solutions. Hence, the SCION-TS solution set is complete and sound.

4.1 Simulations with Input PAGs

Time-series instances of the overlapping networks problem have not previously been studied, so we first determine what is learnable from ideal inputs. We use the same simulation design as in Section 3.1 with 100 connected ground-truth graphs, all Markov order 1, for each setting, though with $\mathcal{N} \in \{8, 10, 15, 20\}$. Input PAGs were then generated by marginalization. We report results for the 92 parameterizations where $\geq 95\%$ of simulations finished; full results available in supplementary material. Results here all use edge direction since it carries important temporal information.

In general, we find that, for the same settings of $\mathcal{N}$ and ground-truth density, the results of SCION-TS were more coherent than SCION: there were fewer graphs in solution sets (Tables 3 & 4), a higher proportion of edges was shared in a given percentage of the output set (Figure 3 left), and

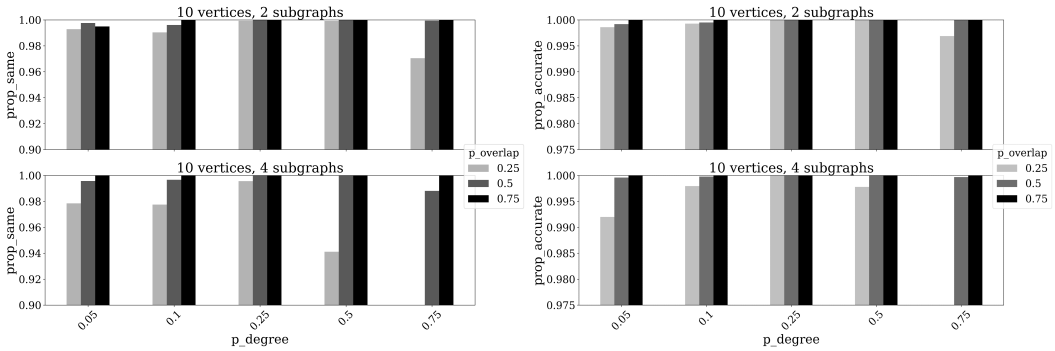


Figure 3: Left: Mean $prop_same$ (75% threshold) for $\mathcal{N} = 10$; Right: Mean $prop_accurate$ (75%).

Table 3: Median number of solutions for 10- (left) and 20-node (right) ground-truth, 2 input PAGs

		$p_{overlap}$		
		0.25	**0.50**	**0.75**
p_{degree}	**0.10**	4.5	1.0	1.0
	0.25	1.0	1.0	1.0
	0.50	31.5	1.0	1.0
	0.75	*	2.0	1.0

		$p_{overlap}$		
		0.25	**0.50**	**0.75**
p_{degree}	**0.10**	1.0	1.0	1.0
	0.25	2.0	1.0	1.0
	0.50	*	3.0	1.0
	0.75	*	*	1.0

more of these shared edges were correct, often reaching 100% (Figure 3 right). These results suggest that time-lag information significantly improves learning by constraining path lengths, even though it adds computational (and representational) complexity. Surprisingly, although SCION-TS runtime scaled with $\mathcal{N}$, the number of output graphs did *not*, perhaps because SCION-TS (implicitly) encodes more pairwise relationships and constraints due to time-lags. At the same time, for given $\mathcal{N}$, increasing ground-truth density and decreasing input overlap does still increase solution set size.

4.2 Simulations with Input Data

We also assess SCION-TS using synthetic data rather than ideal inputs. In this case, p-value pooling is insufficient to ensure a satisfiable SCION-TS input since the estimated lags imply additional constraints on path length (not just path existence). If the initial input is unsatisfiable, then we do a local search to find the smallest number of input edge removals required to achieve a satisfiable solution. A benefit of the ASP approach is that unsatisfiable SCION and SCION-TS problems often fail quickly, making a local search through the input space relatively more tractable. However, we recognize that there may be more efficient or intelligent ways to find "nearby" satisfiable problems.

We again use the stochastic block model to generate our ground-truth graphs, first with two three-node clusters, $\rho_w = 0.5$, $\rho_b \in \{0.05, 0.1\}$, and $\sigma \in \{0.05, 0.1\}$. We generated linear-Gaussian data (5000 timesteps) that were split into two four-variable datasets. We then repeated this with eight-node ground truths with $\rho_w \in \{0.1, 0.25\}$, split into two six-variable input datasets. We use p-value pooling with the LPCMCI (Gerhardus & Runge, 2020) method and a partial-correlation based

Table 4: Mean runtime for SCION-TS for 10- (left) and 20-node (right) ground-truth, 2 input PAGs

		$p_{overlap}$		
		0.25	**0.50**	**0.75**
p_{degree}	**0.10**	0.075	0.129	0.098
	0.25	0.125	0.135	0.087
	0.50	0.218	0.357	0.117
	0.75	*	4.439	0.160

		$p_{overlap}$		
		0.25	**0.50**	**0.75**
p_{degree}	**0.10**	0.669	1.009	0.952
	0.25	3.407	1.461	0.945
	0.50	*	153.352	1.993
	0.75	*	*	9.706

conditional independence test (Fisher, 1924) to learn causal graphs with latent confounders from timeseries data (both implemented in the *tigramite* Python package (Runge, 2018)). LPCMCI allows for bidirected edges and information about absences of paths between nodes. The only permissible lag-0 edges are due to unobserved common causes, so we restrict those edges to be $X_t \leftrightarrow Y_t$.

We ran 100 simulations for each setting (full results in Appendix E). In the 6-node graph simulations, 65–76% of graphs required at least one edge removal to yield a satisfiable problem, and the mean number of edge removals required was between 1.14 and 2.01 (across different parameterizations). In the 8-node simulations, between 84–92% required some edge removal, with mean edge removals between 2.50 and 3.28 to yield a satisfiable result. Accuracy results were similar across simulations: between 73.6% and 80.8% of the shared edge absences or presences were accurate. While unsatisfiable graphs represent *inconsistently* learned inputs, incorrect edges shared in the entire solution represent consistently learned input errors. Overall, SCION-TS is able to extract significant information about the global ground-truth graph, even from actual time-series data.

5 Limitations and Discussion

The outputs of SCION and SCION-TS are provably correct—they return all possible ground-truth graphs consistent with the input—but there are limitations to an ASP-based approach. Contradictory information in inputs, whether due to differences in underlying data distributions or statistical errors in causal discovery, can lead to unsatisfiable inputs. Local search for "nearby" satisfiable inputs (e.g., by structural Hamming distance; Tsamardinos et al. (2006)) can help, as shown in Section 4.2. However, this approach requires more systematic testing of different methods. Of course, even if we find a satisfiable problem, the output may be incorrect due to statistical errors or chance.

A more human-centered limitation arises in interpretation of the output equivalence class, which can contain tens or hundreds of millions of graphs, even given relatively small inputs. Of course, these large output sets are still much smaller than the super-exponential number of $\mathcal{N}$-node DAGs, but they nonetheless might have limited real-world utility. In this paper, we use the proportion of solutions containing a given edge or edge absence as an observable (approximate) confidence metric. Indeed, in our results on input graphs, we found that edges or absences in large proportions of the output set were very likely to be accurate. These "frequent" edges and absences are not, however, usually sufficient to determine a unique output DAG. Typically, additional information or experiments will be required to disambiguate solution set graphs. In those cases, SCION and SCION-TS can help to indicate edges of interest that are likely but not certain to exist, or indicate edges where there is high disagreement within the solution set. SCION and SCION-TS can thus provide information relevant to active learning and experimentation approaches.

Even without leveraging other information, there are potentially other methods or assumptions that could help to deal with the solution set size. For example, suppose two potential ground-truth DAGs $\mathbb{H}_1$ and $\mathbb{H}_2$ are returned by SCION, where the edges in $\mathbb{H}_1$ are a proper subset of those in $\mathbb{H}_2$. We might make a simplicity assumption that justifies a focus on $\mathbb{H}_1$ due to its fewer causal connections. In this fashion, by leveraging additional assumptions or requirements from the data, we can reduce an often very large solution set into more useful constructs for analysis.

In this work, we present SCION and SCION-TS, answer set programming formulations for determining the complete equivalence class of ground-truth graphs with a set of locally learned input graphs. In particular, SCION-TS is the first-ever general algorithm for the overlapping datasets problem in the time series setting. The ASP formulation allows for both greater computational efficiency and a simpler expression of the algorithm and additional constraints, enabling a novel approach to the timeseries overlapping-variables problem. We evaluate both algorithms on simulated input graphs, and show how solution set size and the amount and accuracy of shared edges evolve with varying inputs. We also evaluate both methods given input datasets, and demonstrate potential methods for resolving input inconsistencies. In addition to improving performance, SCION and SCION-TS enable simpler encoding of background knowledge, a method of understanding the global information entailed by a set of locally learned graphs, and could indicate potential targets for experimental intervention when learning real-world causal structures.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods. We have not used generative AI tools to generate synthetic data sets, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, clean and reformat dataset, design or provide feedback on research methodology or experiments, or interpret results; and assistance with translation, and qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to create or edit software code, and summarize or analyse existing literature. We have reviewed all AI-assisted work. AI assistance with implementation and code largely took the form of testable helper functions, rather than the main functions of any simulations, and was verified by authors. (AI assistance was not used in the development of either *clingo* method.) AI review of literature was used as a check for undiscovered parts of the literature, but none of the paper's references were generated with AI assistance. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Appendix A contains proofs for Theorems 1, 2, and 3 as referenced in the main text. Appendices C and E contain results for SCION and SCION-TS simulations on input data, respectively. Appendix D contains full variable descriptions and a summary of the output set for the ESS example outlined in Section 3.2. Full Python code for all simulations of SCION and SCION-TS, in addition to the *clingo* programs themselves, are provided in the supplementary material (with more information available in a README file). The supplement also includes cleaning steps for ESS data, which must be downloaded from the ESS Data Portal at `https://www.europeansocialsurvey.org/data-portal`. Simulation code requires only slight adaptation to a user's computing environment. Full results for SCION and SCION-TS simulations from input graphs, which were too lengthy to conveniently include in the appendix, are also in supplementary material; explanations of column names are provided in the README.

REFERENCES

Mohammadsajad Abavisani, David Danks, and Sergey Plis. Grace-c: Generalized rate agnostic causal estimation via constraints. In *The Eleventh International Conference on Learning Representations*, 2023.

Gerhard Brewka, Thomas Eiter, and Mirosław Truszczyński. Answer set programming at a glance. *Communications of the ACM*, 54(12):92–103, 2011.

Edward C Budd and Daniel B Radner. The obe size distribution series: methods and tentative results for 1964. *The American Economic Review*, 59(2):435–449, 1969.

Thomas Eiter, Giovambattista Ianni, and Thomas Krennwallner. *Answer set programming: A primer*. Springer, 2009.

ESS ERIC. European social survey (ess), round 8 - 2016, 2017. URL `https://ess.sikt.no/en/study/f8e11f55-0c14-4ab3-abde-96d3f14d3c76`.

ESS ERIC. European social survey (ess), round 9 - 2018, 2019. URL `https://ess.sikt.no/en/study/bdc7c350-1029-4cb3-9d5e-53f668b8fa74`.

Ronald A Fisher. On the" probable error" of a coefficient of correlation deduced from a small sample. *Metron*, 1:3–32, 1921.

Ronald Aylmer Fisher. The distribution of the partial correlation coefficient. *Metron*, 3:329–332, 1924.

Ronald Aylmer Fisher. Statistical methods for research workers. 1934.

Martin Gebser, Benjamin Kaufmann, André Neumann, and Torsten Schaub. clasp: A conflict-driven answer set solver. In *Logic Programming and Nonmonotonic Reasoning: 9th International*

Conference, LPNMR 2007, Tempe, AZ, USA, May 15-17, 2007. Proceedings 9, pp. 260–265. Springer, 2007.

Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Multi-shot asp solving with clingo. *Theory and Practice of Logic Programming*, 19(1):27–82, 2019.

Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In *ICLP/SLP*, volume 88, pp. 1070–1080. Cambridge, MA, 1988.

Andreas Gerhardus and Jakob Runge. High-recall causal discovery for autocorrelated time series with latent confounders. *Advances in neural information processing systems*, 33:12615–12625, 2020.

Louis Adedapo Gomez, Jan Claassen, and Samantha Kleinberg. Causal inference for time series datasets with partially overlapping variables. *Journal of Biomedical Informatics*, pp. 104828, 2025.

Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

Biwei Huang, Kun Zhang, Mingming Gong, and Clark Glymour. Causal discovery from multiple data sets with non-identical variable sets. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 10153–10161, 2020.

Antti Hyttinen, Patrik O Hoyer, Frederick Eberhardt, and Matti Jarvisalo. Discovering cyclic causal models with latent variables: A general sat-based procedure. *arXiv preprint arXiv:1309.6836*, 2013.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

Aura Leulescu and Mihaela Agafitei. Statistical matching: a model based approach for data integration. *Eurostat-Methodologies and Working papers*, pp. 10–2, 2013.

Yang Liu, Yan Kang, Chaoping Xing, Tianjian Chen, and Qiang Yang. A secure federated transfer learning framework. *IEEE Intelligent Systems*, 35(4):70–82, 2020.

Victor W Marek and Miroslaw Truszczyński. Stable models and an alternative logic programming paradigm. In *The logic programming paradigm: A 25-year perspective*, pp. 375–398. Springer, 1999.

Benjamin Okner. Constructing a new data base from existing microdata sets: the 1966 merge file. In *Annals of Economic and Social Measurement, Volume 1, Number 3*, pp. 325–362. NBER, 1972.

Gerhard Paass. Statistical match: evaluation of existing procedures and improvements by using additional information. *Microanalytic Simulation Models to Support Social and Financial Policy*, pp. 401–420, 1986.

Kari Rantanen, Antti Hyttinen, and Matti Järvisalo. Learning optimal cyclic causal graphs from interventional data. In *International Conference on Probabilistic Graphical Models*, pp. 365–376. PMLR, 2020.

Willard L Rodgers. An evaluation of statistical matching. *Journal of Business & Economic Statistics*, 2(1):91–102, 1984.

Jakob Runge. Causal network reconstruction from time series: From theoretical assumptions to practical estimation. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(7), 2018.

Richard Scheines, Peter Spirtes, Clark Glymour, Christopher Meek, and Thomas Richardson. The tetrad project: Constraint based aids to causal model specification. *Multivariate Behavioral Research*, 33(1):65–117, 1998.

Shreya Sharma, Chaoping Xing, Yang Liu, and Yan Kang. Secure and efficient federated transfer learning. In *2019 IEEE international conference on big data (Big Data)*, pp. 2569–2576. IEEE, 2019.

Christopher A Sims. Comment on Okner: "Constructing a new data base from existing microdata sets: the 1966 merge file". In *Annals of Economic and Social Measurement, Volume 1, Number 3*, pp. 343–345. NBER, 1972.

AC Singh, H Mantel, M Kinack, and G Rowe. Statistical matching: use of auxiliary information as an alternative to the conditional independence assumption. *Survey Methodology*, 19(1):59–79, 1993.

Kseniya Solovyeva, David Danks, Mohammadsajad Abavisani, and Sergey Plis. Causal learning through deliberate undersampling. In *Conference on Causal Learning and Reasoning*, pp. 518–530. PMLR, 2023.

Dag Sonntag, Matti Järvisalo, José M Peña, and Antti Hyttinen. Learning optimal chain graphs with answer set programming. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pp. 822–831, 2015.

Peter L Spirtes, Christopher Meek, and Thomas S Richardson. Causal inference in the presence of latent variables and selection bias. *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, 1995.

Robert Tillman and Peter Spirtes. Learning equivalence classes of acyclic models with latent and selection variables from multiple datasets with overlapping variables. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 3–15. JMLR Workshop and Conference Proceedings, 2011.

Robert Tillman, David Danks, and Clark Glymour. Integrating locally learned causal structures with overlapping variables. *Advances in Neural Information Processing Systems*, 21, 2008.

Sofia Triantafillou and Ioannis Tsamardinos. Constraint-based causal discovery from multiple interventions over overlapping variable sets. *Journal of Machine Learning Research*, 16(66):2147–2205, 2015. URL `http://jmlr.org/papers/v16/triantafillou15a.html`.

Sofia Triantafillou, Ioannis Tsamardinos, and Ioannis Tollis. Learning causal structure from overlapping variable sets. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 860–867. JMLR Workshop and Conference Proceedings, 2010.

Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65:31–78, 2006.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

Li-Chun Zhang. On proxy variables and categorical data fusion. *Journal of Official Statistics*, 31 (4):783–807, 2015.

Yujia Zheng, Biwei Huang, Wei Chen, Joseph Ramsey, Mingming Gong, Ruichu Cai, Shohei Shimizu, Peter Spirtes, and Kun Zhang. Causal-learn: Causal discovery in python. *Journal of Machine Learning Research*, 25(60):1–8, 2024.

A Proofs of Theorems

Proof of Soundness. Suppose X and Y are d-separated given $\mathbf{Z}$ in some $\mathbb{G}_i$, but d-connected in some output $\mathbb{H}_i$. This implies that there is a path between X and Y in $\mathbb{H}_i$ that is active given $\mathbf{Z}$. X and Y are not adjacent in $\mathbb{G}_i$, and so (by line 22) the output graph d-connection cannot be a directed path or common cause. The only remaining possibility is that some variable $R \in \mathbf{Z}$ is a descendant of a collider in $\mathbb{H}_i$ on a path between X and Y. This implies, however, that $\mathbb{H}_i$ includes paths from

X to R and Y to R that are active given $\mathbf{Z} \backslash R$. However, this implies (per lines 10-11) that each of these paths corresponds to a sequence of edges in $\mathbb{G}_i$ that contradict the known d-separation in $\mathbb{G}_i$.

Now suppose that X and Y are d-connected given $\mathbf{Z}$. Line 23 specifies that if an edge exists between two nodes X and Y in input graph T, then the property `directed(X,Y,T)` is true. Per lines 10 and 11, `directed(X,Y,T)` holds true only when there is an edge from X to Y in the output, or when the solution includes multiple edges from X to Y consisting of intermediate nodes that were not observed in graph T. This means that any pair of nodes connected by an edge in an input T will be connected either by a single edge, or by a directed path of nodes that were not included in T. This, in turn, entails the necessary d-connection relation. □

Proof of Completeness. In order to show completeness, we must show that no d-separations or d-connections present in the input graph are unnecessarily removed from the output set $\mathcal{H}$. All edge removals in line 4 are necessary to translate d-separations from the inputs, as is the acyclicity constraint in line 5. Remaining edge removals only occur in line 20 by removing bidirected edges $X \leftrightarrow Y$ and retaining the relevant d-connections by creating directed paths to X and Y from the unobserved common cause, or in line 11 to replace a directed edge $X \rightarrow Y$ with a previously unobserved path of edges $X \rightarrow Z \rightarrow Y$. Because *clingo* outputs the entire set of solution graphs matching the given constraints, and because none of the changes specified by these constraints would preclude such an output $\mathbb{H}_i$ from the solution set, SCION is complete for the problem. □

Proof of Theorem 3. First, if nodes X and Y are connected in some input graph $\mathbb{G}_i$ by a directed edge with lag $L = 1$, then every output graph $\mathbb{H}_i$ will contain the edge $X \rightarrow Y$. This is given by Line 3 of Listing 2. This line also gives that if there is not an edge between X and Y with lag 1 in the true output, then there will not be an output edge between these nodes.

If nodes X and Y are not connected in any input graph $\mathbb{G}_i \in \mathcal{G}$, then they will not be connected in any output graph $\mathbb{H}_i \in \mathcal{H}$. This is also given by Line 3.

If nodes X and Y are connected in some input graph $\mathbb{G}_i \in \mathcal{G}$ by a directed edge with lag $L > 1$, then there will be a path $X \rightarrow \overset{L-1}{\cdots} \rightarrow Y$ in every output graph $\mathbb{H}_i \in \mathcal{H}$, where $\overset{L-1}{\cdots}$ represents $L - 1$ nodes not contained in $\mathbb{G}_i$. This is given by Line 14. Then, Line 15 gives that if there is not such a connection between X and Y in the input, then there will not be a directed path consisting of unobserved nodes of this length in any output set.

Suppose nodes X and Y are connected in some input graph $\mathbb{G}_i \in \mathcal{G}$ by a bidirected edge with lag $L > 1$ (indicating a causal connection between X at time t and Y and time $t + L$). Then, there will be a pair of paths $Z \rightarrow \overset{L_1}{\cdots} \rightarrow X$ and $Z \rightarrow \overset{L_2}{\cdots} \rightarrow Y$ in every output graph $\mathbb{H}_i \in \mathcal{H}$, where $\overset{L_1}{\cdots}$ and $\overset{L_2}{\cdots}$ represent paths of nodes (with lengths L_2 and L_1, respectively) not contained in $\mathbb{G}_i$, such that $L = L_2 - L_1$. This is given by Lines 17 through 19: if there is a bidirected edge between two comeasured variables in a given graph, then there must be a `common_cause` (i.e., a pair of unobserved `directed` paths to an unobserved common ancestor that differ in length by L). Then, Lines 21 through 23 require that if there is not such a bidirected connection in the input between comeasured variables, then a corresponding set of unobserved paths to a common ancestor cannot exist.

Since these conditions summarize all possible relationships between variables in the inputs, and since the *clingo* program both preserves the correct causal relationships from the inputs and precludes the existence of the corresponding causal structures when these input patterns are not present, Listing 2 is a direct encoding of the SCION-TS problem. Since Clingo is a complete solver, and all of the possible graphs are considered, this is sufficient to show that the solution from SCION-TS is both complete and sound. □

B ION BASELINE RUNTIME COMPARISON

The following table reports runtime results for ION on some 8-node graphs, with a comparison to runtimes for the corresponding simulations of SCION, as described in Section 3.1. In the following table, "nodes" refers to the number of vertices in the ground-truth, p_d and p_o to settings of density and overlap, and s to the number of subgraphs the ground truth was split into, all as described in

Section 3.1. "Finished" refers to the number of graphs where either ION or SCION returned a result within the four-hour timeout, with "mean" and "median" representing the mean and median runtimes of all graphs that completed. Since in our main simulations, we only report these figures for parameterizations where at least 95/100 simulations finished, we denote runtime figures from ION parameterizations where fewer graphs completed with an asterisk to indicate that these figures are unlikely to be representative.

				ION runtime			**SCION runtime**		
nodes	p_d	p_o	s	**finished**	**mean**	**median**	**finished**	**mean**	**median**
8	0.10	0.25	2	13	3279.475*	1402.001*	100	0.053	0.038
8	0.10	0.25	4	1	305.948*	305.948*	100	1.185	0.047
8	0.10	0.50	2	77	227.272*	1.120*	100	0.028	0.020
8	0.10	0.50	4	72	418.143*	1.299*	100	0.021	0.016
8	0.10	0.75	2	99	2.613	0.240	100	0.020	0.013
8	0.10	0.75	4	100	0.273	0.223	100	0.021	0.013
8	0.25	0.50	2	43	864.489*	19.396*	100	0.024	0.017
8	0.25	0.50	4	39	1390.198*	272.619*	100	0.064	0.018
8	0.25	0.75	2	99	73.843	1.622	100	0.019	0.013
8	0.25	0.75	4	100	8.389	0.356	100	0.021	0.014
8	0.50	0.75	2	72	1915.309*	141.860*	100	0.022	0.013
8	0.50	0.75	4	87	684.075*	28.403*	100	0.019	0.012

C SCION SYNTHETIC DATA SIMULATIONS

- **nodes**: number of nodes in the ground-truth
- **in**: within-group edge density in the stochastic block model (SBM)
- **out**: between-group edge density in the stochastic block model (SBM)
- **error**: standard deviation of Gaussian error term in generating dataset from ground-truth DAG
- O**: overlap**: Number of nodes in common between the two subgraphs
- **time**: median runtime of SCION on satisfiable input problem
- **solutions**: median number of graphs in SCION solution set on satisfiable problem, rounded to the nearest whole
- S_p: Identical to $prop_same$ and $prop_same_unmerged$ in main results: proportion of edges or edge absences in solution set which are shared between $p\%$ of solution graphs, both merging and not merging solution edges of opposite orientations.
- A_p: Identical to $prop_accurate$ in main results: of edges or edge absences in solution set which are shared between $p\%$ of solution graphs, which proportion are found in the ground truth?

All inputs refer to 8-node ground-truth graphs as described in Section 3.2. Both tables below refer to the same sets of simulations, but report edge frequency analysis under two different methods (merging and not merging edges between the same nodes of opposite orientation).

C.1 MERGING EDGES OF OPPOSITE ORIENTATION

in	**out**	**error**	O	**time**	**solutions**	S_{75}	S_{90}	S_{100}	A_{75}	A_{90}	A_{100}
0.25	0.05	0.05	3	1.974	221131	0.508	0.457	0.457	0.975	0.978	0.978
0.25	0.05	0.05	4	0.341	45118	0.582	0.54	0.537	0.977	0.978	0.978
0.25	0.05	0.1	3	1.379	173878	0.508	0.461	0.46	0.967	0.967	0.967
0.25	0.05	0.1	4	0.362	46545	0.586	0.542	0.538	0.97	0.968	0.968
0.25	0.1	0.05	3	2.377	301932	0.495	0.447	0.445	0.96	0.961	0.962
0.25	0.1	0.05	4	0.575	67528	0.568	0.53	0.527	0.963	0.962	0.962
0.25	0.1	0.1	3	1.954	255106	0.503	0.45	0.447	0.96	0.964	0.963
0.25	0.1	0.1	4	0.644	67319	0.563	0.521	0.517	0.967	0.967	0.967
0.25	0.25	0.05	3	7.698	955094	0.458	0.412	0.409	0.907	0.918	0.918
0.25	0.25	0.05	4	1.824	229607	0.519	0.481	0.476	0.899	0.906	0.906
0.25	0.25	0.1	4	1.34	174443	0.518	0.486	0.484	0.905	0.907	0.907
0.5	0.05	0.05	4	0.541	71041	0.574	0.524	0.519	0.946	0.941	0.941
0.5	0.05	0.1	4	0.59	80654	0.557	0.518	0.516	0.938	0.934	0.934
0.5	0.1	0.05	4	0.758	100718	0.541	0.501	0.5	0.906	0.905	0.904
0.5	0.1	0.1	4	0.969	129098	0.541	0.504	0.502	0.903	0.902	0.901
0.5	0.25	0.05	4	2.281	275229	0.508	0.477	0.476	0.804	0.8	0.8
0.5	0.25	0.1	4	1.667	201225	0.516	0.481	0.479	0.818	0.819	0.819

C.2 NOT MERGING EDGES OF OPPOSITE ORIENTATION

in	**out**	**error**	O	S_{75}	S_{90}	S_{100}	A_{75}	A_{90}	A_{100}
0.25	0.05	0.05	3	0.736	0.558	0.457	0.937	0.972	0.989
0.25	0.05	0.05	4	0.746	0.606	0.537	0.945	0.982	0.989
0.25	0.05	0.1	3	0.733	0.555	0.46	0.932	0.971	0.983
0.25	0.05	0.1	4	0.744	0.611	0.538	0.931	0.971	0.984
0.25	0.1	0.05	3	0.722	0.534	0.445	0.918	0.965	0.981
0.25	0.1	0.05	4	0.731	0.598	0.527	0.936	0.974	0.981
0.25	0.1	0.1	3	0.72	0.543	0.447	0.925	0.969	0.982
0.25	0.1	0.1	4	0.735	0.594	0.517	0.933	0.973	0.984
0.25	0.25	0.05	3	0.675	0.506	0.409	0.895	0.941	0.959
0.25	0.25	0.05	4	0.697	0.553	0.476	0.893	0.945	0.953
0.25	0.25	0.1	4	0.714	0.558	0.484	0.899	0.945	0.953
0.5	0.05	0.05	4	0.746	0.588	0.519	0.92	0.964	0.97
0.5	0.05	0.1	4	0.742	0.589	0.516	0.913	0.957	0.967
0.5	0.1	0.05	4	0.728	0.581	0.5	0.908	0.942	0.952
0.5	0.1	0.1	4	0.72	0.574	0.502	0.901	0.941	0.951
0.5	0.25	0.05	4	0.699	0.55	0.476	0.856	0.898	0.9
0.5	0.25	0.1	4	0.712	0.55	0.479	0.857	0.904	0.91

D FULL ESS VARIABLE DESCRIPTIONS AND RESULTS

Variable	Description	Survey round(s)
sbbsntx	Social benefits/services cost businesses too much in taxes/charges	2016 only
basinc	Against or in favour of a basic income scheme	2016 only
ctzcntr	Are you a citizen of [country]?	Both surveys
lrscale	Placement on left right scale	Both surveys
brncntr	Born in country	Both surveys
iphlppl	Important to help people and care for others well-being	Both surveys
ifrjob	Compared other people in country, fair chance get job I seek	2018 only
netifr	Your net [pay/pensions/social benefits] is unfairly low, fair, or unfairly high	2018 only

Node 1	Node 2	# of solutions with $1 \to 2$ edge	Proportion of agreement on edge or absence
sbbsntx	basinc	66258	0.703
sbbsntx	ctzcntr	0	1.000
sbbsntx	lrscale	33588	0.849
sbbsntx	brncntr	0	1.000
sbbsntx	iphlppl	0	1.000
sbbsntx	ifrjob	33012	0.852
sbbsntx	netifr	71676	0.679
basinc	sbbsntx	66258	0.703
basinc	ctzcntr	0	1.000
basinc	lrscale	33588	0.849
basinc	brncntr	0	1.000
basinc	iphlppl	0	1.000
basinc	ifrjob	33012	0.852
basinc	netifr	71676	0.679
ctzcntr	sbbsntx	0	1.000
ctzcntr	basinc	0	1.000
ctzcntr	lrscale	0	1.000
ctzcntr	brncntr	74391	0.667
ctzcntr	iphlppl	0	1.000
ctzcntr	ifrjob	0	1.000
ctzcntr	netifr	0	1.000
lrscale	sbbsntx	72144	0.677
lrscale	basinc	72144	0.677
lrscale	ctzcntr	0	1.000
lrscale	brncntr	0	1.000
lrscale	iphlppl	71208	0.681
lrscale	ifrjob	0	1.000
lrscale	netifr	77940	0.651
brncntr	sbbsntx	0	1.000
brncntr	basinc	0	1.000
brncntr	ctzcntr	74391	0.667
brncntr	lrscale	0	1.000
brncntr	iphlppl	74391	0.667
brncntr	ifrjob	0	1.000
brncntr	netifr	0	1.000
iphlppl	sbbsntx	0	1.000
iphlppl	basinc	0	1.000
iphlppl	ctzcntr	0	1.000
iphlppl	lrscale	75276	0.663

iphlppl	brncntr	74391	0.667
iphlppl	ifrjob	0	1.000
iphlppl	netifr	59346	0.734
ifrjob	sbbsntx	69336	0.689
ifrjob	basinc	69336	0.689
ifrjob	ctzcntr	0	1.000
ifrjob	lrscale	0	1.000
ifrjob	brncntr	0	1.000
ifrjob	iphlppl	0	1.000
ifrjob	netifr	76986	0.655
netifr	sbbsntx	49896	0.776
netifr	basinc	49896	0.776
netifr	ctzcntr	0	1.000
netifr	lrscale	60885	0.727
netifr	brncntr	0	1.000
netifr	iphlppl	11304	0.949
netifr	ifrjob	59382	0.734

E SCION-TS SYNTHETIC DATA SIMULATIONS

- **in**: within-group edge density in the stochastic block model (SBM)
- **out**: between-group edge density in the stochastic block model (SBM)
- **error**: standard deviation of Gaussian error term in generating dataset from ground-truth DAG
- **overlap**: Number of nodes in common between the two subgraphs
- **runtime**: median runtime of SCION-TS on satisfiable input problem
- **solutions**: median number of graphs in SCION-TS solution set on satisfiable problem
- **# edges**: mean number of edges that need to be removed from the SCION-TS input in order to yield a satisfiable problem
- **% graphs**: proportion of graphs for which any edge removal is required to yield a satisfiable problem
- S_p: identical to $prop_same$ in main results: proportion of (directed) edges or edge absences in solution set which are shared between $p\%$ of solution graphs
- A_p: identical to $prop_accurate$ in main results: of (directed) edges or edge absences in solution set which are shared between $p\%$ of solution graphs, which proportion are found in the ground truth?

The following two tables refer to the same set of simulations, with the first table reporting information about the runtime, solution set, and edge removal, and the second reporting information about the edge frequency and accuracy. Note that in the timeseries case, we do not merge edges in opposite directions for any results.

nodes	in	out	error	overlap	runtime	solutions	# edges	% graphs
6	0.5	0.05	0.05	2	0.012	4	1.398	0.724
6	0.5	0.05	0.05	4	0.01	2	1.869	0.758
6	0.5	0.05	0.1	2	0.012	4	1.137	0.653
6	0.5	0.05	0.1	4	0.01	4	1.66	0.711
6	0.5	0.1	0.05	2	0.011	4	1.32	0.763
6	0.5	0.1	0.05	4	0.01	4	2.01	0.753
6	0.5	0.1	0.1	4	0.01	2.5	1.68	0.69
8	0.1	0.05	0.05	4	0.011	3	2.701	0.856
8	0.1	0.05	0.1	4	0.011	3	2.711	0.918
8	0.1	0.1	0.05	4	0.011	2	2.5	0.844
8	0.1	0.1	0.1	4	0.011	3	2.832	0.884
8	0.25	0.05	0.05	4	0.011	2	3.278	0.918
8	0.25	0.05	0.1	4	0.011	2	2.663	0.878

nodes	in	out	error	overlap	S_{75}	S_{90}	S_{100}	A_{75}	A_{90}	A_{100}
6	0.5	0.05	0.05	2	0.974	0.918	0.916	0.747	0.736	0.736
6	0.5	0.05	0.05	4	0.939	0.939	0.939	0.76	0.76	0.76
6	0.5	0.05	0.1	2	0.97	0.907	0.903	0.756	0.744	0.743
6	0.5	0.05	0.1	4	0.934	0.934	0.934	0.757	0.757	0.757
6	0.5	0.1	0.05	2	0.969	0.913	0.91	0.751	0.742	0.742
6	0.5	0.1	0.05	4	0.927	0.926	0.926	0.741	0.741	0.741
6	0.5	0.1	0.1	4	0.943	0.942	0.942	0.739	0.739	0.739
8	0.1	0.05	0.05	4	0.974	0.958	0.952	0.814	0.812	0.812
8	0.1	0.05	0.1	4	0.978	0.955	0.951	0.811	0.809	0.808
8	0.1	0.1	0.05	4	0.982	0.966	0.963	0.806	0.804	0.804
8	0.1	0.1	0.1	4	0.974	0.958	0.956	0.807	0.807	0.806
8	0.25	0.05	0.05	4	0.978	0.966	0.964	0.79	0.789	0.788
8	0.25	0.05	0.1	4	0.982	0.967	0.966	0.797	0.795	0.795