A Appendix

Contents

Algorithm 1 Training Algorithm

procedure TRAINING(SCM Black Box Distribution D, with M nodes and N categories)
  Let i an index from 0 to $M-1$

  for I iterations, or until convergence, **do**
    for F functional parameter training steps **do** ▷ **Phase 1**
      $X \sim D$
      $C \sim \mathrm{Ber}(\sigma(\gamma))$
      $L = -\log P(X|C\,;\theta)$
      $\theta_{t+1} \leftarrow \mathrm{Adam}(\theta_t, \nabla_\theta L)$

    for Q interventions **do**
      ▷ **Phase 2**
      `I_N` $\leftarrow$ `randint(0,` $M-1$`)` ▷ Black Box, hidden
      $D_{\text{int}} := D$ with intervention on node `I_N` ▷ Black Box, hidden

      if predicting intervention **then** ▷ Phase 2 Prediction
        $L_i \leftarrow 0 \quad \forall i$
        for N_P prediction steps **do**
          $X \sim D_{\text{int}}$
          for C_P configurations **do**
            $C \sim \mathrm{Ber}(\sigma(\gamma))$
            $L_i \leftarrow L_i - \log P_i(X|C_i;\theta)\ \forall i$
        `I_N` $\leftarrow$ `argmax(`L_i`)`

      `gammagrads, logregrets = [], []` ▷ Phase 2 Scoring
      for N_S scoring steps **do**
        $X \sim D_{\text{int}}$
        `gammagrad, logregret = 0, 0`
        for C_S configurations **do**
          $C \sim \mathrm{Ber}(\sigma(\gamma))$
          $L_i = -\log P_i(X|C_i;\theta) \quad \forall i$
          `gammagrad +=` $\sigma(\gamma) - C$ ▷ Collect $\sigma(\gamma) - C$ for Equation 3
          `logregret +=` $\sum_{i \neq \texttt{I\_N}} L_i$ ▷ Collect $\mathcal{L}^{(k)}_{C,i}(X)$ for Equation 3
        `gammagrads.append(gammagrad)`
        `logregrets.append(logregret)`

      ▷ **Phase 3**
      $g_{ij} = \dfrac{\sum_k (\sigma(\gamma_{ij}) - c^{(k)}_{ij}) \mathcal{L}^{(k)}_{C,i}(X)}{\sum_k \mathcal{L}^{(k)}_{C,i}(X)}$ ▷ Gradient Estimator, Equation 3
      $g \leftarrow g + \nabla_\gamma \left(\lambda_{\text{sparse}}\, L_{\text{sparse}}(\gamma) + \lambda_{\text{DAG}}\, L_{\text{DAG}}(\gamma)\right)$ ▷ Regularizers
      $\gamma_{t+1} \leftarrow \mathrm{Adam}(\gamma_t, g)$

A.1 Training Algorithm

Algorithm 1 shows the pseudocode of the method for training SDI. Typical values for the loop trip counts are found in §A.11.

A.2 Preliminaries

Interventions. In a purely-observational setting, it is known that causal graphs can be distinguished only up to a Markov equivalence class. In order to identify the true causal graph intervention data is needed (Eberhardt et al., 2012). Several types of common *interventions* may be available (Eaton & Murphy, 2007b). These are: *No intervention:* only observational data is obtained from the ground truth causal model. *Hard/perfect:* the value of a single or several variables is fixed and then ancestral sampling is performed on the other variables. *Soft/imperfect:* the conditional distribution of the variable on which the intervention is performed is changed. *Uncertain:* the learner is not sure of which variable exactly the intervention affected directly. Here we make use of soft interventions for several reasons: First, they include hard interventions as a limiting case and hence are more general. Second, in many real-world scenarios, it is more difficult to perform a hard intervention compared to a soft one. We also deal with a special case of uncertain interventions,

where the variable selected for intervention is random and unknown. We call these *unidentified* or *unknown* interventions.

Intervention setup. For our experiments, the groundtruth models of the synthetic datasets are modeled by neural networks as described in section A.6. Each neural network models the relationship of the causal parents and a variable. We perform our intervention by first randomly selecting which variable to intervene on, then soft-intervening on it. The selected variable is sampled from a uniform distribution. The soft intervention is a reinitialization of its neural network's parameters.

Causal sufficiency. The inability to distinguish which causal graph, within a Markov equivalence class, is the correct one in the purely-observational setting is called the *identifiability problem*. In our setting, all variables are observed (there are no latent confounders) and all interventions are random and independent. Hence, within our setting, if the interventions are known, then the true causal graph is always identifiable in principle (Eberhardt et al., 2012; Heinze-Deml et al., 2018a). We also consider here situations where a single variable is randomly selected and intervened upon with a soft or imprecise intervention, its identity is unknown and must be inferred. In this case, there is no theoretical guarantee that the causal graph is identifiable. However, there is existing work (Peters et al., 2016) that handles this scenario and the proposed method is also proven to work empirically.

Faithfulness. It is possible for causally-related variables to be probabilitistically independent purely by happenstance, such as when causal effects along multiple paths cancel out. This is called *unfaithfulness*. We assume that *faithfulness* holds, since the γ gradient estimate is extracted from shifts in probability distributions. However, because of the "soft" nature of our interventions and their infinite variety, it would be exceedingly unlikely for cancellation-related unfaithfulness to persist throughout the causal-learning procedure.

A.3 Experimental setup

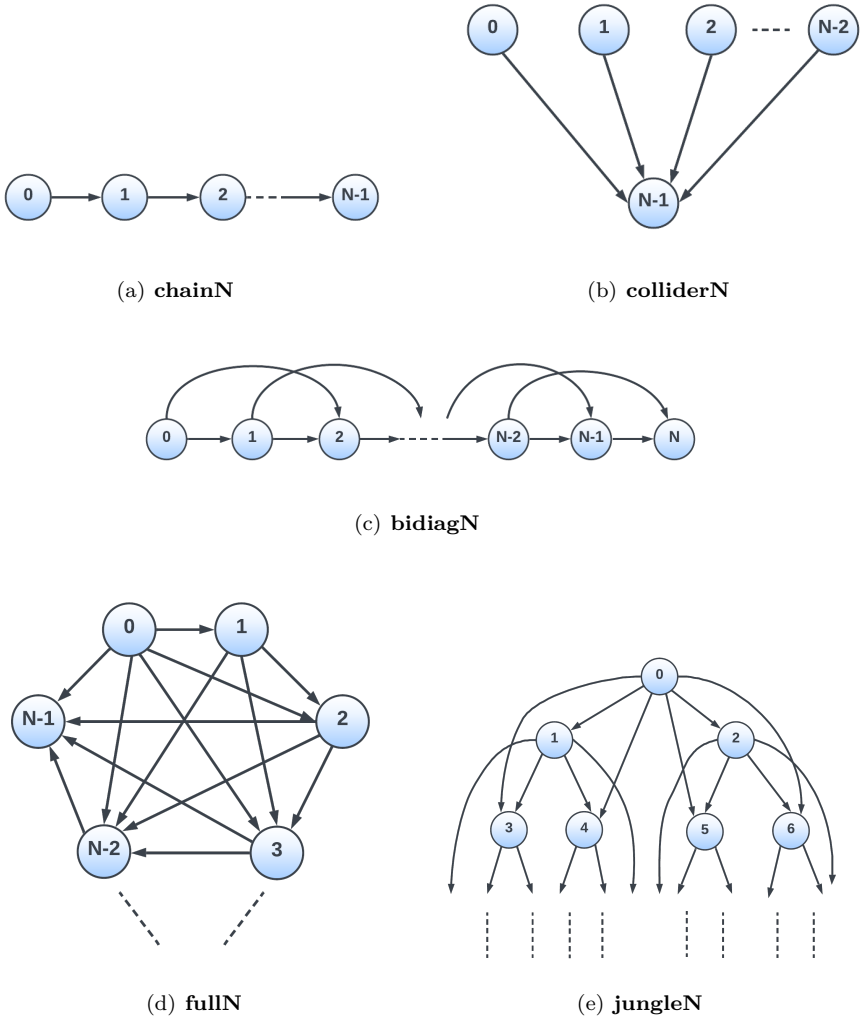
For all datasets, the weight parameters for the learned model is initialized randomly. In order to not bias the structural parameters, all $\sigma(\gamma)$ are initialized to 0.5 in the beginning of training. Details of hyperparameters of the learner model are described in Section A.5. The experimental setup for the groundtruth model for the synthetic data can be found in Section A.6 and the details for the real world data are described in Section A.7.

A.4 Model setup

Wwe model the M variables in the graph using M independent MLPs, each possesses an input layer of $M \times N$ neurons (for M one-hot vectors of length N each), a single hidden layer chosen arbitrarily to have $\max(4M, 4N)$ neurons with a LeakyReLU activation of slope 0.1, and a linear output layer of N neurons representing the unnormalized log-probabilities of each category (a softmax then recovers the conditional probabilities from these logits). To force f_i to rely exclusively on the direct ancestor set $pa(i, C)$ under adjacency matrix C (See Eqn. 3), the one-hot input vector X_j for variable X_i's MLP is masked by the Boolean element c_{ij}. The functional parameters of the MLP are the set $\theta = \{\texttt{WO}_{ihjn}, \texttt{BO}_{ih}, \texttt{W1}_{inh}, \texttt{B1}_{in}\}$.An example of the multi-MLP architecture with M=3 categorical variables of N=2 categories is shown in Figure 2.

A.5 Hyperparameters

Both the functional and structural parameters are optimized using the Adam optimizer (Kingma & Ba, 2014). We use a learning rate of $5e-2$ with alpha of 0.9 for the functional parameters, and we use a learning rate of $5e-3$ with alpha of 0.1 for the structural parameters. We perform 5 runs of each experiment with random seeds $1-5$ and error bars are plotted for graphs up to size 13 in Figure 10. We use a batch size of 256. The L1 norm regularizer is set to 0.1 and the DAG regularizer is set to 0.5 for all experiments. We study the effect of DAG and sparsity penalties in Section 6.6. For each γ update step, we sample 25 structural



(a) **chainN** (b) **colliderN**

(c) **bidiagN**

(d) **fullN** (e) **jungleN**

Figure 6: Figures for various synthetic graphs. `chain`, `collider`, `bidiagonal`, `full` and `jungle` graph.

configurations from the current γ. In all experiments, we use 100 batches from the interventional distribution to predict the intervened node.

Learner model. All experiments on the synthetic graphs of size 3-8 use the same hyperparameters. Both the functional and structural parameters are optimized using the Adam optimizer (Kingma & Ba, 2014). We use a learning rate of $5e-2$ with alpha of 0.9 for the functional parameters, and we use a learning rate of $5e-3$ with alpha of 0.1 for the structural parameters. We perform 5 runs of each experiment with random seeds $1-5$ and error bars are plotted for various graphs from size 3 to 8 in Figure 10. We use a batch size of 256. The L1 norm regularizer is set to 0.1 and the DAG regularizer is set to 0.5 for all experiments. For each γ update step, we sample 25 structural configurations from the current γ. In all experiments, we use 100 batches from the interventional distribution to predict the intervened node.

A.6 Synthetic data

We study a variety of SCMs with different ground-truth edge structures γ. Our selection of synthetic graphs explores various extremes in the space of DAGs, stress-testing SDI. The `chain` graphs are the sparsest connected graphs possible, and are relatively easy to learn. The `bidiag` graphs are extensions of `chain`

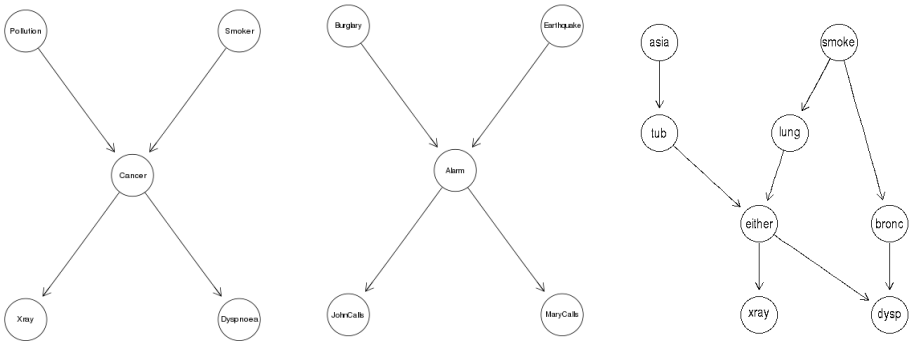


Figure 7: Left to right: Ground Truth SCM for Cancer, Groundtruth SCM for Earthquake, Groundtruth SCM for Asia.

where there are 2-hops as well as single hops between nodes, doubling the number of edges and creating a meshed chain of forks and colliders. The `jungle` graphs are binary-tree-like graphs, but with each node connected directly to its grandparent in the tree as well. Half the nodes in a `jungle` graph are leaves, and the out-degree is up to 6. The `collider` graphs deliberately collide independent $M-1$ ancestors into the last node; They stress maximum in-degree. Lastly, the `full` graphs are the maximally dense DAGs. All nodes are direct parents of all nodes below them in the topological order. The maximum in- and out-degree are both $M-1$. These graphs are depicted in Figure 6.

A.6.1 Synthetic data results

The model can recover correctly all synthetic graphs with 10 variables or less, as shown in Figure 10 and Table 1. For graphs larger than 10 variables, the model found it more challenging to recover the denser graphs (e.g. `fullM`), as shown in Table 1. Plots of the training curves showing average cross entropy (CE) and Area-Under-Curve(AUC/AUCROC) for edge probabilities of the learned graph against the ground-truth graph for synthetic SCMs with 3-13 variables are available in Figure 10.

A.7 BnLearn data repository

The repo contains many datasets with various sizes and structures modeling different variables. We evaluate the proposed method on 3 of the datasets in the repo, namely the Earthquake (Korb & Nicholson, 2010), Cancer (Korb & Nicholson, 2010) and Asia (Lauritzen & Spiegelhalter, 1988) datasets. The ground-truth model structure for the Cancer (Korb & Nicholson, 2010) and Earthquake (Korb & Nicholson, 2010) datasets are shown in Figure 7. Note that even though the structure for the 2 datasets seems to be the same, the conditional probability tables (CPTs) for these 2 datasets are very different and hence results in different structured causal models (SCMs) for the 2 datasets.

Method	`Asia`	`chain8`	`jungle8`	`collider7`	`collider8`	`full8`
(Zheng et al., 2018)	14	24	14	11	18	21
(Yu et al., 2019)	10	7	12	6	7	25
(Heinze-Deml et al., 2018b)	8	7	12	6	7	28
(Peters et al., 2016)	5	3	8	4	2	16
(Eaton & Murphy, 2007a)	0	0	0	7	7	1
sdis	0	0	0	0	0	0

Table 5: **Baseline comparisons:** Hamming distance (lower is better) for learned and ground-truth edges on various graphs from both synthetic and real datasets, compared to (Peters et al., 2016), (Heinze-Deml et al., 2018b), (Eaton & Murphy, 2007b), (Yu et al., 2019) and (Zheng et al., 2018). The proposed SDI is run on random seeds $1-5$ and we pick the worst performing model out of the random seeds in the table.

Figure 8: Learned edges at three different stages of training. **Left**: `chain4` (chain graph with 4 variables). **Right**: `full4` (tournament graph with 4 variables).

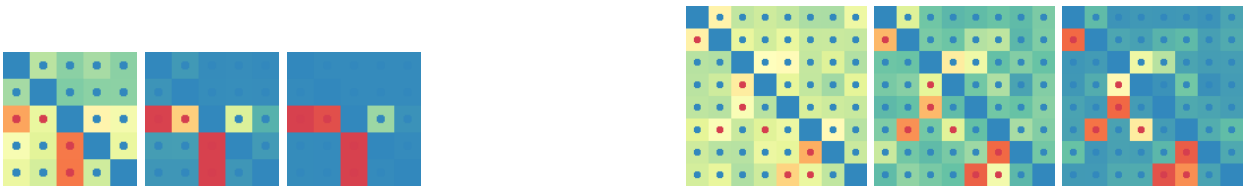


Figure 9: **Left:** Earthquake: Learned edges at three different stages of training. **Right:** Asia: Learned edges at three different stages of training.

Given that some of the CPTs contain very unlikely events, we have found it necessary to add a *temperature* parameter in order to make them more frequent. The near-ground-truth MLP model's logit outputs are divided by the temperature before being used for sampling. Temperatures above 1 result in more uniform distributions for all causal variables; Temperatures below 1 result in less uniform, sharper distributions that peak around the most likely value. We find empirically that a temperaure of about 2 is required for our BnLearn benchmarks.

A.8 Comparisons to other methods

We compare to 5 other methods. The full comparison between SDIs and other methods on various graphs can be found in Table 1.

One of these methods, DAG-GNN Yu et al. (2019), outputs 3 graphs based on different criteria: best mean square error (MSE), best negative loglikelihood (NLL) and best evidence lower bound (ELBO). We report performance of all outputs of DAG-GNN Yu et al. (2019) in Table 6, and the best one is selected for Table 1.

	sdi	Best MSE	Best NLL	Best Elbo
`Asia`	0	10	10	13
`chain8`	0	7	7	7
`jungle8`	0	12	12	13
`collider7`	0	6	6	6
`collider8`	0	8	8	7
`full8`	0	27	25	27

Table 6: **Baseline comparisons:** Hamming distance (lower is better) for learned and ground-truth edges on `Asia` and various synthetic graphs. compared to DAG-GNN Yu et al. (2019). DAG-GNN outputs 3 graphs according to different criterion. We show results on all outputs in this table and we show the best performing result in Table 1.

A.9 Sparsity of Ground-Truth Graph

We evaluated the performance of SDI on graphs of various size and sparsity to better understand the performance of the model. We evaluated the proposed model on 4 representative types of graphs in increasing order of density. They are the `chain`, `jungle`, `bidiag` and `full` graphs. As shown in the results in figure 11, for graphs of size 5 or smaller, there is almost no difference in the final results in terms of variance and sample complexity. However, as the graphs gets larger (than 6), the denser graphs (`full` graphs) gets progressively

	Chain						Jungle						Full					
	3	4	5	6	7	8	3	4	5	6	7	8	3	4	5	6	7	8
ldag=0.5, lsparse=0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ldag=0.5, *lsparse*=0.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ldag=0., *lsparse*=0.1	0	0	0	**1**	0	0	0	0	0	**1**	**3**	**1**	0	0	0	0	**1**	**6**

Table 7: **Regularizer:** SDI performance measured by Hamming distance to the ground-truth graph. Comparisons are between SDIs with different regularizer settings for different graphs. Our default setting is ldag = 0.5, lsparse = 0.1, with ldag the DAG regularization strength and lsparse the sparsity regularization strength. As shown in the table, SDIs is not very sensitive to different regularizer settings. Tasks with non-zero Hamming distance (errors) are in bold.

more difficult to learn compared to the sparser graphs (`chain`, `jungle` and `bidiag`). The models learned for denser graphs have higher complexity, higher variance and slightly worse results.

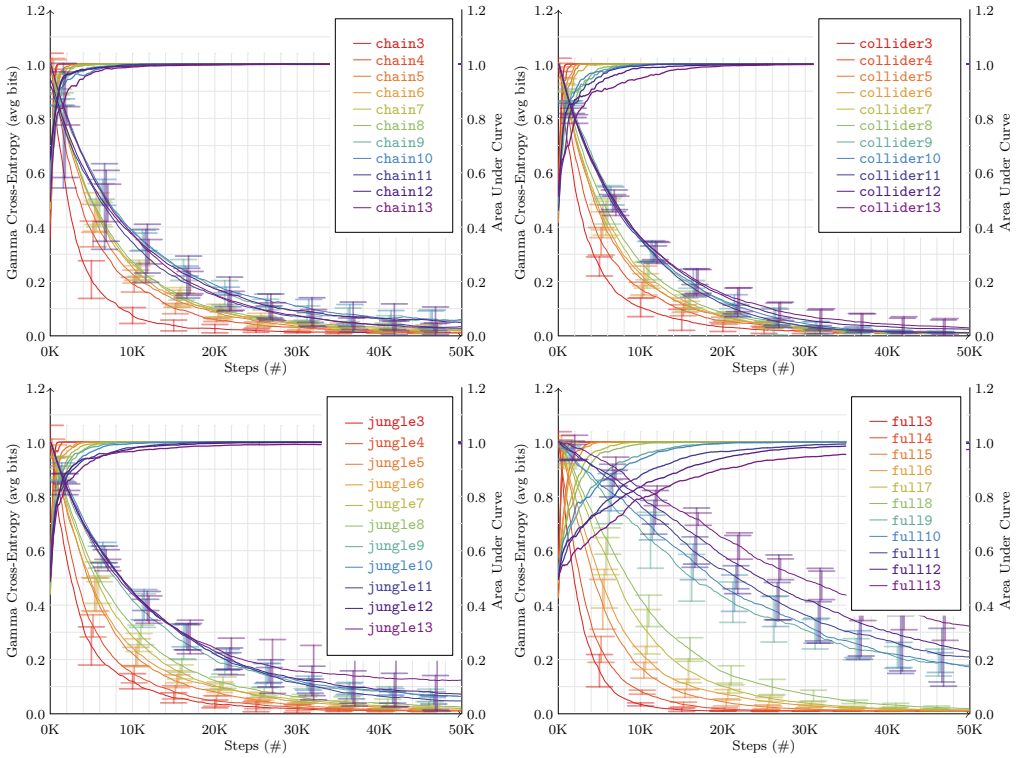


Figure 10: Cross entropy (CE) and Area-Under-Curve (AUC/AUROC) for edge probabilities of learned graph against ground-truth for synthetic SCMs. Error bars represent $\pm 1\sigma$ over PRNG seeds 1-5. **Left to right, up to down**: `chainM`,`jungleM`,`fullM`,$M = 3 \ldots 8$ ($9 \ldots 13$ in Appendix A.6.1). Graphs (3-13 variables) all learn perfectly with AUROC reaching 1.0. However, denser graphs (fullM) take longer to converge.

A.10 Predicting interventions

In Phase 2, we score graph configurations based on how well they fit the interventional data. We find that it is necessary to avoid disturbing the learned parameters of intervened variables, and to ignore its contribution to the total negative log-likelihood of the sample. Intuitively, this is because, having been intervened upon, that variable should be taken as a given. It should especially not be interpreted as a poorly-learned variable requiring a tuning of its functional parameters, because those functional parameters were not responsible for the value of that variable; the extrinsic intervention was.

Since an intervened variable is likely to be unusually poorly predicted, we heuristically determine that the most poorly predicted variable is the intervention variable. We then zero out its contribution to the log-likelihood of the sample and block gradient into its functional parameters.

Figure 5 illustrates the necessity of this process. When using the prediction heuristic, the training curve closely tracks training with ground-truth knowledge of the identity of the intervention. If no prediction is made, or a random prediction is made, training proceeds much more slowly, or fails entirely.

A.11 Sample complexity

Our method relies on sampling of configurations and data in Phases 1 and 2. We present here the breakdown of the sample complexity. Let

- I be the number of iterations of the method, *(typical: 500-2000)*
- B the number of samples per batch, *(typical: 256)*
- F the number of functional parameter training iterations in Phase 1, *(typical: 10000)*
- Q the number of interventions performed in Phase 2, *(typical: 100)*
- N_P the number of data batches for prediction, *(typical: 100)*
- C_P the number of graph configurations drawn per prediction data batch, *(typical: 10)*
- N_S the number of data batches for scoring, *(typical: 10)*
- C_S the number of graph configurations drawn per scoring data batch. *(typical: 20-30)*

Then the total number of interventions performed, and configurations and samples drawn, over an entire run are:

$$\text{Interventions} = IQ = \gamma \text{ updates} \tag{4}$$

$$\text{Samples} = I(\underbrace{F}_{\text{Phase 1}} + \underbrace{Q(N_P + N_S)}_{\text{Phase 2}})B \tag{5}$$

$$\text{Configurations} = I(\underbrace{F}_{\text{Phase 1}} + \underbrace{Q(C_P N_P + C_S N_S)}_{\text{Phase 2}}) \tag{6}$$

Because of the multiplicative effect of these factors, the number of data samples required can quickly spiral out of control. For typical values, as many as $500 \times 10000 \times 256 = 1.28\text{e}9$ observational and $500 \times 100 \times (100 + 10) \times 256 = 1.408\text{e}9$ interventional samples are required. To alleviate this problem slightly, we limit the number of samples generated for each intervention; This limit is usually 500-2000.

A.12 Runtimes

The runtime for different methods on 10-node graphs is as follows. Our proposed method SDI, takes between 4 to 5 hours for training. In comparison, DAG-GNN (Yu et al., 2019) requires between 2 to 3 hours, and DAG-Notears takes around half an hour. The training time for ICP (Peters et al., 2016) is approximately 4 hours, and the non-linear ICP (Heinze-Deml et al., 2018a) also takes about 4 hours. Additionally, Eaton & Murphy (2007a) requires approximately 12 hours for training.

A.13 Effect of regularization

Importance of sparsity regularizer. We use a $L1$ regularizer on the structure parameters γ to encourage a sparse representation of edges in the causal graph. In order to better understand the effect of the $L1$ regularizer, we conducted ablation studies on the $L1$ regularizer. It seems that the regularizer has an small effect on rate of converges and that the model converges faster with the regularizer, This is shown in Figure 12. However, this does not seem to affect the final value the model converges to, this is shown in Table 7.

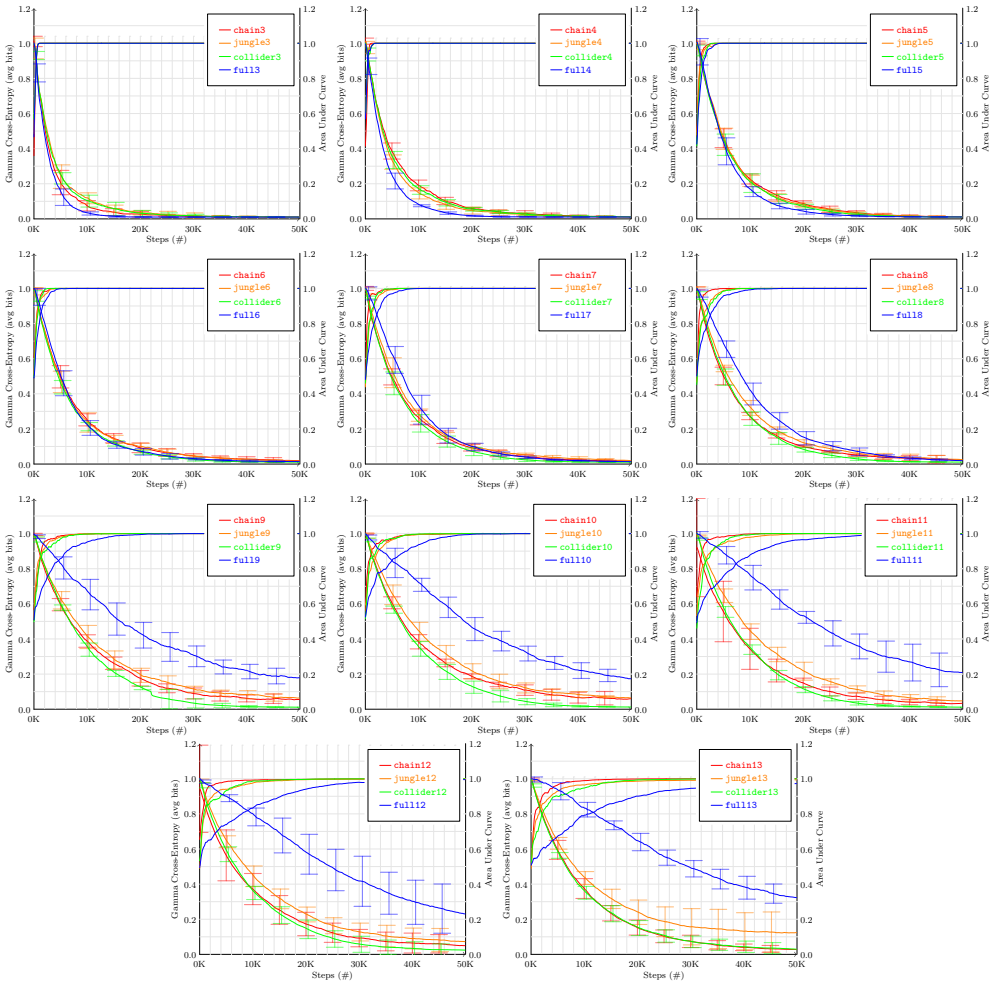


Figure 11: **Left to right, top to bottom** Average cross-entropy loss of edge beliefs $\sigma(\gamma)$ and Area-Under-Curve throughout training for the synthetic graphs `chainN`, `jungleN`, `colliderN` and `fullN`, N=3-13, grouped by graph size. Error bars represent $\pm 1\sigma$ over PRNG seeds 1-5.

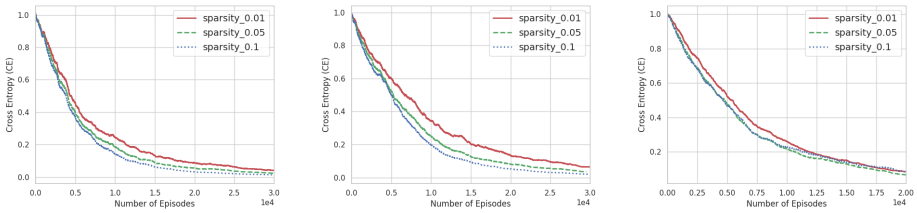


Figure 12: **Effect of sparsity ($lsparse$) regularizer :** On 5 variable, 6 variable and 8 variable Nodes

Importance of DAG regularizer. We use an acyclic regularizer to discourage length-2 cycles in the learned model. We found that for small models (≤ 5 variables), the acyclic regularizer helps with faster convergence, without improving significantly the final cross-entropy. This is illustrated for the 3-variable graphs in Figure 13. However, for graphs larger than 5 variables, the acyclic regularizer starts playing an important role in encouraging the model to learn the correct structure. This is shown in the ablation study in Table 7.

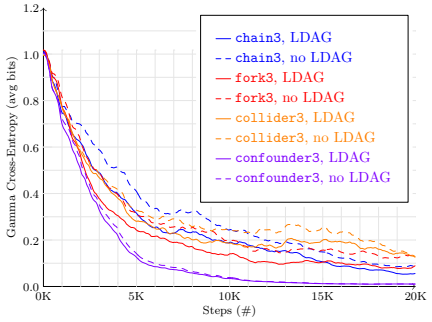


Figure 13: Ablations study results on all possible 3 variable graphs. Graphs show the cross-entropy loss on learned vs ground-truth edges over training time. Comparisons of model trained with and without DAG regularizer (L_{DAG}), showing that DAG regularizer helps convergence.

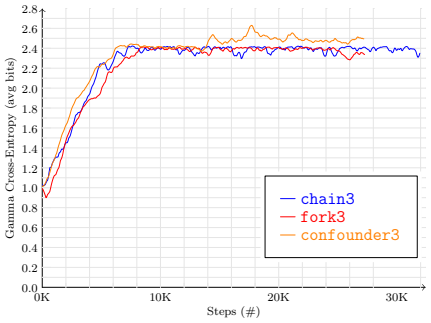


Figure 14: Edge CE loss for 3-variable graphs with no dropout when training functional parameters, showing the importance of this dropout.

A.14 Importance of dropout

To train the functional parameters on an observational distribution, one would need sampling adjacency matrices. One may be tempted to make these "complete directed graph" (all-ones except for a zero diagonal), to give the MLP maximum freedom to learn any potential causal relations itself. We demonstrate that functional parameter training cannot be carried out this way, and that it is necessary to "drop out" each edge (with probability of the current γ value in our experiments) during pre-training of the conditional distributions of the SCM. We attempt to recover the previously-recoverable graphs `chain3`, `fork3` and `confounder3` without dropout, but fail to do so, as shown in Figure 14.

	sdi	**Eaton & Murphy (2007b)**
`Asia`	0	0
`chain8`	0	0
`jungle8`	0	0
`collider7`	0	7
`collider8`	0.0	7
`full8`	0.0	1

Table 8: **Comparisons:** Structured hamming distance (SHD) on learned and ground-truth edges on `asia` and various synthetic graphs. Eaton & Murphy (2007b) can not scale to larger variables graphs as shown in Table 1, hence, we compare to the largest graph that (Eaton & Murphy, 2007b) can scale up to. SDI is compared to (Eaton & Murphy, 2007b) for `collider7`, `collider8` and `full8`, (Eaton & Murphy, 2007a) asserts with 100% confidence a no-edge where there is one (false negative). For comparisons with all other methods 1.

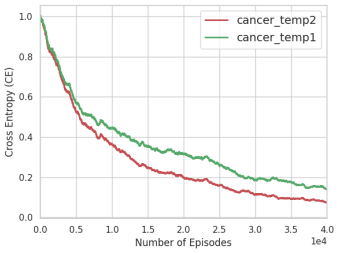


Figure 15: Cross-entropy for edge probability between learned and ground-truth SCM for Cancer at varying temperatures.

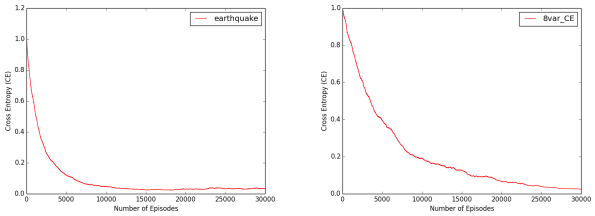


Figure 16: Cross-entropy for edge probability between learned and ground-truth SCM. **Left**: The Earthquake dataset with 6 variables. **Right**: The Asia dataset with 8 variables