

Deep learning of causal structures in high dimensions under data limitations

In the format provided by the
authors and unedited

Contents

1	Overview and main characteristics	2
1.1	Additional comments on D ² CL	2
1.2	Related Work	2
2	Problem Statement	3
2.1	Notation	4
2.2	Problem Statement	4
2.3	Summary of learning scheme	4
3	Experimental Setup	5
3.1	Overview of Datasets	5
3.1.1	Gold-standard simulated benchmark data	5
3.1.2	Yeast Gene Deletion Experiments	6
3.1.3	CRISPR-based interventional data in human cells	8
3.2	Baseline Comparisons	8
4	Additional Results	10
4.1	Gold-standard simulated benchmark data: Direct causal effects	10
4.2	Gold-standard simulated benchmark data: Total causal effects	10
4.3	Gold-standard simulated benchmark data: Scalability experiment with $p=50,000$	10
4.4	Identifying Causal Directions	11
4.5	Multiplicative Noise	11
4.6	Out-of-distribution tests	12
4.7	Yeast Gene Deletion Experiments	14
4.8	Wall-clock times	15
5	Data Generation: Pseudo-Code	16
5.1	SEM Data Generation	17
5.2	Functions	17

1 Overview and main characteristics

1.1 Additional comments on D²CL

D²CL uses an end-to-end neural approach to learn causal networks from a combination of empirical data and prior causal knowledge. The architecture consists of two “towers”, a deep convolutional network (CNN) and a graph neural network (GNN), that process different representations of the inputs X and Π . The towers are respectively aimed at leveraging distributional patterns in the empirical data X and graphical/structural features (as exploited in link prediction tasks). The entire model is trained end-to-end. The general D²CL workflow and its application to biomedical problems are summarized in Figure 1 of the manuscript.

As noted in the introduction of the main text, D²CL operates within a causal risk framework and differs from a number of existing causal structure learning schemes (e.g. those based on causal graphical models). The proposed approach has the following main characteristics:

- D²CL focuses on causal learning for real-world, high-dimensional problems with many nodes (potentially thousands) but limited data availability and lack of gold-standard simulation engines. This is motivated in particular by high-dimensional biomedical problems. Acyclicity (of the directed graphs to be learned) is not assumed, nor is availability of any standard factorization of the joint probability distribution.
- For D²CL it is not required that samples in the data matrix X are drawn from a single distribution. Samples can be drawn from, e.g., a mix of observational, and interventional distributions and the causal characteristics of these regimes (e.g. which node(s) or latents were intervened upon) need not be known in advance. This is a common setup for real-world data and in particular for emerging experimental designs in biology (see examples below).
- During the training phase, pattern and feature detectors are trained in a supervised fashion. These learned patterns are statistical representations of certain aspects of underlying causal processes (see Sec. 2) and possess causal semantics via the input labels, in a similar sense to the way learned feature maps in image processing and object recognition implicitly capture image semantics.
- For inference, the trained network can be used to output a directed causal relation between *any* pair of variables and thus, D²CL is able to provide a global graph (i.e. over all nodes) which could possibly include cycles. Thus, the model is trained on partial knowledge of some cause-effect relations (prior causal knowledge Π) but seeks to generalize to the complete problem.
- Inference of causal direction is achieved via a combination of causal knowledge encoded in supervision labels and the fact that the NNs used are *not* rotation-invariant (since if an image feature is rotated, in general this will not match the expected input of a filter trained on non-rotated input). Although lack of rotation invariance is often not desirable in image processing, the proposed learner exploits this to break symmetries. Hence, the output adjacency matrices are *not* symmetric in general.

Thus, D²CL is aimed at scientific use cases in which empirical data and causal prior knowledge are available, but where the problem dimension may be very large. It leverages a neural approach within a causal risk framework to allow efficient scaling and a highly nonparametric, demonstrably effective overall scheme.

1.2 Related Work

The rich body of work in learning causal structures includes, among other methods, PC [1], LiNGAM [2], IDA [3], GIES [4], RFCI [5] and ICP [6]. The proposed D²CL differs fundamentally from classical causal structure learning methods rooted in causal graphical models, as it focuses on in a sense “directly” learning the (directed) causal status of edges. This framing means that D²CL can effectively learn novel causal edges (as seen in the experiments reported) and scales well to larger problems, but cannot on its own provide richer output, such as full interventional distributions. The learning framework upon which D²CL is based was introduced in MRCL [7]; the key difference is that while MRCL is based on a classical semi-supervised

Table S1: Comparison of algorithms

Method	Causal Model	Method Type	Link Function	Latent Confounders	Demonstrated scalability	Variable Type	Sample size used	Intervention Type
SDI	generative	continuous constraint	linear non-linear	No	48	categorical		single variable intervention, unknown target, soft intervention
DCDI	generative	continuous constraint	linear non-linear	No	100	categorical continuous	100000 samples, interventions on all nodes	regime interventions, soft stochastic interventions, hard stochastic interventions, known/unknown targets
DCD-FG	generative	continuous constraint	linear non-linear	No	1000	continuous	>50000 observational and interventional samples	regime interventions, hard stochastic interventions, known/unknown targets
ENCO	generative	continuous constraint	linear non-linear	No	1000 (categorical)	categorical continuous	100000 observational samples, 4096 interventional samples per node	regime interventions, soft stochastic interventions, hard stochastic interventions, full intervention information
CSiVA	discriminative	meta-learning	linear non-linear	Possibly	80	categorical continuous	40000 graphs, 1500 interventional and observational samples each graph	soft and hard interventions, single variable interventions
DDCL	discriminative	supervised learning	linear non-linear	Possibly	50000	continuous	>1000 observational and interventional samples	hard deterministic interventions, hard stochastic interventions, single variable interventions, unknown targets

manifold learning scheme, D²CL uses neural networks in a supervised fashion and can scale to much larger problems that is possible using MRCL. [8] studied causal risk functions in the context of assessment of existing learners; D²CL leverages the notion of causal risk to propose a new learner.

There have been important recent efforts at the intersection of neural networks and causality. In particular, scaling causal structure learning to large problems has been facilitated by reformulation as a continuous optimization problem [9] and recent neural approaches, e.g. SDI [10], DCDI [11], DCD-FG [12], or ENCO [13] have further built on this idea. A key difference with respect to D²CL is that the latter is not based on a DAG framework at all, but pursues “direct” mapping of data inputs to graph outputs. This is an approach shared by the recently proposed CSiVA [14]. The key difference is that while CSiVA uses a meta-learning scheme based on large-scale synthetic data, D²CL is based on supervised learning using data from a specific system of interest (for example a biological system). Both CSiVA and D²CL do not make very specific assumptions on the observational or interventional data. Furthermore, neither algorithm assumes acyclicity. The fact that D²CL is formulated as supervised learning task while CSiVA is designed as a meta-learning task is reflected in data availability assumptions. During training, CSiVA uses large numbers of gold-standard causal systems (via simulation engines) and learns to identify features that generalize across systems. D²CL is trained and applied on a single system and focuses on generalizing to new interventions/edges in the same system.

Thus, D²CL is suitable for problems, such as in biology, where we do not have easy access to truly gold-standard simulation engines (for large-scale meta-learning) and where a new system of interest (e.g. diseased cells or a different cell type or organism) may differ considerably in variable set, causal and distributional patterns. By training in a context-specific fashion, D²CL is able to successfully learn structures in a range of scenarios. Furthermore, D²CL is demonstrably scalable to large numbers of variables (we show examples ranging up to $p = 50000$ nodes) and does not require very large sample data inputs (relative to related methods). A comparison to other neural methods for causal learning appears in the Table.

2 Problem Statement

This section provides methodological details; to maintain readability and in the interests of providing a self-contained description, parts of what follows repeat material in the main text but with extended description of the notation, problem statement and causal interpretation.

2.1 Notation

Observed variables with index set $V = \{1, \dots, p\}$ are denoted $X_1, \dots, X_p$. The variables will be identified with vertices in a directed graph G whose vertex and edge sets are denoted $V(G), E(G)$, respectively.

We use linear indexing of variable pairs to aid formulation as a machine learning problem. Specifically, an ordered pair $(i, j) \in V \times V$ has an associated linear index $k \in \mathcal{K} = \{1, \dots, K\}$, where K is the total number of variable pairs of interest. Where useful we make the mapping explicit, denoting the linear index corresponding to a pair (i, j) as $k(i, j)$ and the variable pair corresponding to a linear index k as $(i(k), j(k))$. The linear indices of pairs whose causal relationships are *unknown* and of interest are $\mathcal{U} \subset \mathcal{K}$ and those pairs known in advance via input knowledge Π are $\mathcal{T}(\Pi) \subset \mathcal{K}$. In all experiments $\mathcal{T}(\Pi)$ and $\mathcal{U}$ are disjoint, i.e., no prior causal information is available on the pairs $\mathcal{U}$ of interest.

2.2 Problem Statement

We focus on the setting in which available inputs are

- (I1) Empirical data: an $n \times p$ data matrix X whose columns correspond to variables $X_1, \dots, X_p$.
- (I2) Causal background knowledge Π providing information on a subset $\mathcal{T}(\Pi) \subset \mathcal{K}$ of causal relationships.

For (I2), we assume that information is available concerning the causal status of a subset of variable pairs. That is, for some variable pairs (X_i, X_j) the correct binary indicator G_{ij}^* , e.g., presence/absence of an edge in the target graphical object, is provided as an input. In terms of linear indexing, these can be viewed as available “labels” of causal status for the pairs $\mathcal{T}(\Pi) \subset \mathcal{K}$. No specific assumption is made on the data X , but in line with our focus on generalizing to unseen causal relationships, it is assumed that it does *not* contain interventional data corresponding to the pairs in $\mathcal{U}$. Furthermore, in all experiments, not only are the sets $\mathcal{T}$ and $\mathcal{U}$ disjoint, but we enforce the stronger requirement that $u \in \mathcal{U} \implies \nexists j : k(i(u), j) \in \mathcal{T}$, i.e. all interventions on which models are tested are entirely novel, i.e. unrepresented in the inputs to the learner.

Thus, the learning task can be formulated as follows: Given the inputs (I1) and (I2), the goal is to estimate for each ordered pair of variables (X_i, X_j) with unknown causal relationship whether or not X_i has a causal influence on X_j .

2.3 Summary of learning scheme

With the notation above, our goal is to learn a graph whose nodes correspond to the variables $X_1, \dots, X_p$ and edges represent causal relationships. To this end, we train a parameterized network F_θ , i.e., a nonlinear function F with a set of unknown, trainable parameters θ . This is possible since we know for each pair $k \in \mathcal{T}$ the causal status G_{ij}^* based on input information Π . The architecture we use as F_θ is detailed below, but for now assume this has been specified. Then, given the data X and the training labels $Y_k = G_{i(k), j(k)}^*$ for all pairs $k \in \mathcal{T}(\Pi)$, we train the set of parameters $\hat{\theta}(X, \Pi)$ under a loss that is supervised by the (causal) labels Y_k .

At this stage, the trained network $F_{\hat{\theta}(X, \Pi)}$ allows assignment of causal status to *any* pair since it gives an estimate of the entire graph including those pairs whose causal status was unknown. The output is given by:

$$\hat{G}_{ij}(X, \Pi) = \begin{cases} F_{\hat{\theta}(X, \Pi)}(i, j; X) & \text{if } k(i, j) \notin \mathcal{T}(\Pi) \\ Y_{k(i, j)}(\Pi) & \text{otherwise} \end{cases} \quad (1)$$

where (i, j) are ordered variable pairs. Note that the overall estimate depends solely on the data X and causal information Π . By default, no change is made for pairs $\mathcal{T}$ whose status was known at the outset.

Eigenmann et al. [8] studied causal notions of risk based on loss functions of the form $L(\hat{G}, G^*)$ that compare a graph estimate $\hat{G}$ with ground-truth G^* . Given a causal estimator $\hat{G} = F_G(X)$, the theoretical risk is

$$R(F_G) = \mathbb{E}[L(G^*, F_G(X))]. \quad (2)$$

There are many different types of loss functions L that could in principle be used, depending on the specific application and the nature of the problem being solved. In our setting, we consider a classification-type loss on the variable pairs k , where the causal status of known pairs $\mathcal{T}(\Pi)$ provides the training “labels”. Therefore, the general risk estimator takes the form of a cross-entropy loss over ordered variable pairs k . This penalizes the model for assigning high probabilities to incorrect edges and for assigning low probabilities to correct edges. We further augment the loss function by standard terms that, for instance, prevent exploding weights. This notion of risk takes into account the finite sample size and provides an estimate of the method’s performance based on the available data. Furthermore, the problem-specific nature of the risk estimator acknowledges that a particular method may perform well in certain settings, but not in others, depending on the specific characteristics of the system being studied, in alignment with our system-specific approach.

3 Experimental Setup

In general terms, we use data X and causal information Π (in real problems, derived from interventional experiments) to train the learners and then test the output against entirely unseen interventional data. The data are strictly split in the sense that (i) in all experiments the pairs $\mathcal{U}$ on which the model output is tested are disjoint from those pairs $\mathcal{T}$ whose causal relationships are provided as training inputs and (ii) no data used to define the true causal relationships against which the model output is tested appear in inputs to the models.

3.1 Overview of Datasets

3.1.1 Gold-standard simulated benchmark data

Structural Equation Model based on Directed Acyclic Graph: Given the set of variables $X_1, \dots, X_p$, our chosen SEM is a set of p functions, each corresponding to one of the observed variables. Without loss of generality, the SEM comprises equations

$$X_i = f_{X_i}(Pa(X_i), U_{X_i}), \quad i = 1, \dots, p,$$

where $Pa(X_i)$ is the set of parents for node i and the U_{X_i} ’s are exogeneous noise variables assumed jointly independent but arbitrarily distributed. The structure of this SEM follows a directed acyclic graph (DAG) and a variety of synthetic datasets is generated for empirical investigation. The functions f are

1. linear: $f(X_i) = \mathbf{a}Pa(X_i) + b + U_{X_i}$
2. MLP-tanh: $f(X_i) = \mathbf{W}_2 \tanh(\mathbf{W}_1 Pa(X_i) + \mathbf{b}_1) + b_2 + U_{X_i}$
3. MLP-leaky ReLU: $f(X_i) = \mathbf{W}_2 f_{l.ReLU}(\mathbf{W}_1 Pa(X_i) + \mathbf{b}_1) + b_2 + U_{X_i}$
4. tangent hyperbolic: $f(X_i) = \tanh(\text{norm}(Pa(X_i)) + U_{X_i})$
5. leaky ReLU: $f(X_i) = f_{l.ReLU}(\text{norm}(Pa(X_i)) + U_{X_i})$
6. polynomial of order three: $f(X_i) = a_1 Pa(X_i) + a_2 (Pa(X_i))^2 + a_3 (Pa(X_i))^3 + b + U_{X_i}$

Additionally, we consider measurement noise at different signal-to-noise ratios varying from $SNR = 10$ to $SNR = 0.1$. Python-like pseudocode is given in Section 5 below.

Definition of ground truth graph G^* In a first series of experiments, we sought to investigate **direct causal effects**. That is, for a pair of nodes i, j , the corresponding edge $i \rightarrow j$ the ground truth adjacency matrix is set to one, if and only if node i is in the parent set of node j , i.e. $i \in Pa_{G^*}(j)$. In other words, $G_{ij} = 1$, if node i appears directly on the right hand side in the equation for node j . The full set of results appears in Table S2.

The causal graph G^* in the above examples encodes direct causal effects. However, in many real-world examples, interest focuses also on indirect effects, that may be mediated by other nodes. For example, if node A has a direct effect on B , and B on C , intervention on A may change C , even though A does not itself appear in the equation for C . To study the ability to identify such indirect effects, we next tested the various methods on the task of learning indirect edges. This was done in the same way as above, but with the inputs Π being indirect edges and output tested against a (true, gold-standard) indirect graph. That is, in this example, if A causes B and B causes C , the gold-standard graph (and, where relevant, causal inputs in Π) would have an additional edge $A \rightarrow C$ (to capture the causal, but indirect, effect of A on C). Other aspects were as for the direct problems above. In a second series of experiments, we report performance metrics for causal relationships of this kind. This simulated dataset differs from the previous one in the sense that the ground truth graph includes indirect edges of this kind. We used the same set of transition functions as before. (We note that this experimental setup is arguably closer to the real-world biological data, where effects may be indirect in this sense.) The entire set of results is presented in Table S4

3.1.2 Yeast Gene Deletion Experiments

We use data from yeast gene deletion experiments [15], which have previously been used for causal learning [6, 16, 7]. To define causal status, we follow the approach of [7], considering changes under intervention relative to the observational distribution. The data is a collection of $n^{int} = 1479$ interventional samples and $n^{obs} = 153$ observational data points each containing measured gene expression levels for a total of $p = 5535$ genes (after preprocessing). The interventional samples are from gene knockout experiments (each carried out for a specific target gene) while observational samples stem from experiments with no such intervention.

To ensure that models are tested on entirely unseen interventions, we split the data such that the set of interventions against which model output is tested is entirely disjoint from any model input. Below, the number of interventions whose effects are available to the learner via Π is denoted m .

To define causal status, we follow the approach of [7], considering changes under intervention relative to the observational distribution for the same gene. Specifically, if c_{ij}^{int} is the expression level of gene j observed under intervention on gene i , for any variable pair $(i, j) \in V \times V$ we say that i has a causal effect on j if and only if $z_{ij} = |c_{ij}^{int} - m_j^{obs}| / s_j^{obs} > \tau$, where m_j^{obs} is the median level of gene j (in the observational data); s_j^{obs} is the corresponding inter-quartile range (IQR) and τ is a threshold (set to 5 in the experiments). In other words, if G^* is the gold-standard $p \times p$ adjacency matrix, we have $G_{ij}^* = 1 \iff z_{ij} > \tau$. This approach focuses attention on strong causal effects and ensures that the notion of “change under intervention” is appropriate to the scale of each gene.

Note that in the yeast example causal effects may be indirect and our goal in the analysis is to learn a directed graph G with nodes corresponding to p observed genes and edges $(i, j) \in E(G)$ that represent (possibly indirect) causal influences, i.e. existence of a directed causal path from X_i to X_j (possibly via latent variables). Such edges are scientifically interesting as they are relatively amenable to experimental verification as noted in [17, 18]. Cycles can arise in systems biology (see e.g. [19]) and we do not enforce acyclicity in this example (see [20] and references therein, for discussion of cyclic causality). A fuller discussion of the causal interpretation of laboratory experiments is beyond the scope of this paper, but relevant work includes [21, 20, 22] and we direct the interested reader to these references for further discussion.

We conducted the following experiments:

- **Varying amount of causal input.** In a first set of experiments, we seek to investigate the influence of varying the amount of causal input provided to the learners. More precisely, we vary the number of interventions m whose effects are available to the learner. (In terms of the notation above $m = |\{i :$

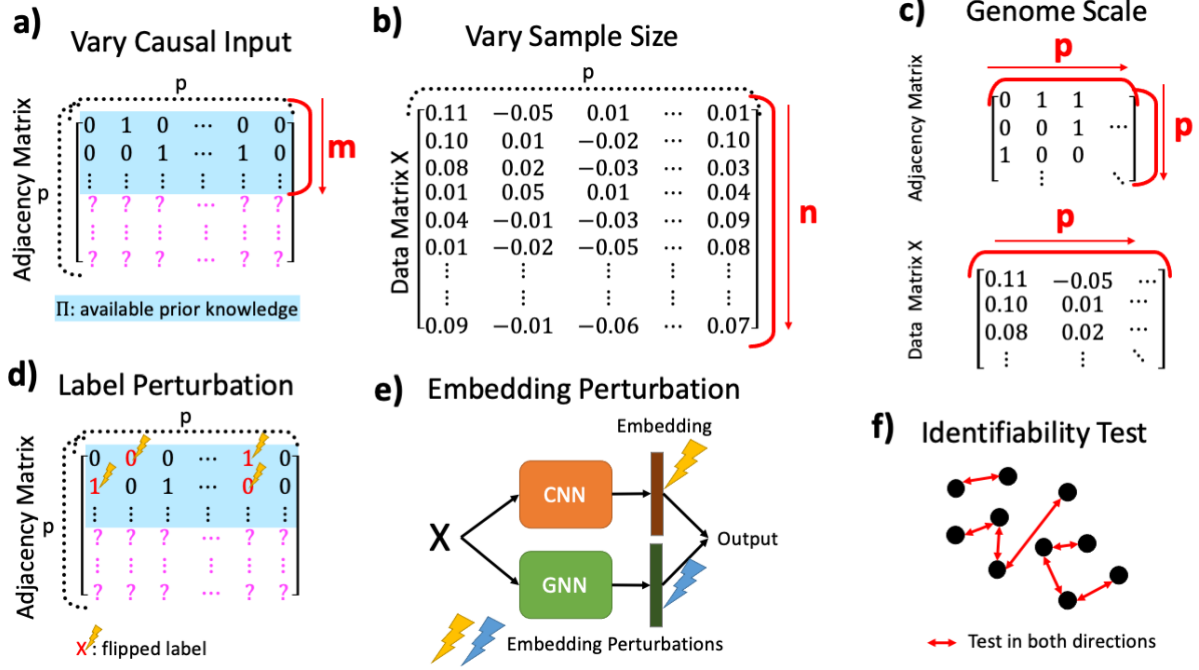


Figure S1: Visual illustration of changed configurations in evaluation of yeast gene deletion data (see Text for details): (a) Varying amount of causal input m , (b) Varying sample size n , (c) Scaling to full yeast genome, (d) Label perturbation, (e) Embedding perturbation, (f) Causal direction analysis

$\exists j, k(i, j) \in \mathcal{T}(\Pi)\}$. Results are shown in Figure 4a-c.

- **Varying sample size.** Here, we vary the sample size n of the data matrix X . (The matrix X never contains interventions on variables V and its sample size can therefore be varied separately from the number m of training interventions or the dimension $p = |V|$.) Results are given in Figure 4d-f.
- **Scaling to full yeast genome.** This experiment is a higher dimensional example with $p = 5535$ genes aimed at investigating performance in a larger problem. Since none of the baseline algorithms can scale to such a high-dimensional space, Figure 4g-k only contains evaluation results for D²CL.
- **Label perturbation.** To study sensitivity, we introduced errors into Π ; this was done by perturbing 10% of the training labels (i.e. labelling causal pairs as non-causal and vice versa) at the outset. Results are presented in Figure 5a.
- **Embedding perturbation.** In this experiment, we intentionally perturb the independent embeddings of the two towers in the forward pass of D²CL. In general, the embedding of either the CNN or the GNN tower is modified right before the fusion layer to test the impact of a failing tower on the overall performance. The experiment contains four different embedding modifications: (i) We set the complete embedding of one tower to zero and thereby, erase all information of this tower. In the other cases we apply strong zero-mean Gaussian noise with varying scale to the embedding, (ii) $\sigma = 1.0$, (iii) $\sigma = 2.0$, and (iv) $\sigma = 5.0$. Sufigure 5b illustrates the results under embedding perturbations.
- **Causal direction analysis.** To empirically study the performance to recover directed and asymmetric causal relations, we constructed test datasets as follows: For each causal edge $k \rightarrow l$ of the test set, we also include the *reverse* direction $l \rightarrow k$. Intuitively, this means that any learner that predicts undirected causal links would achieve an AUC score of 0.5 since the prediction of a causal edge for $k \rightarrow l$ entails the prediction of edge $l \rightarrow k$, which, consequently, is a false positive. Table S7 summarizes the empirical results and Figure 5c presents a low-dimensional representation of the feature maps of the CNN tower right ahead of the embedding fusion.

In the first and second series of experiments, we limit the total number of variables to $p = 1000$ due to computational considerations for the baseline algorithms (not for D²CL, which as we show further below can practically scale to very large problems). Since this is only a subset of the entire yeast genome (a total of 5535 genes were included in the full set of data of the third line of experiments), in these experiments (many) latent variables (genes not included in the dataset in addition to other biological variables, such as proteins, not measured in the data) are present by design.

3.1.3 CRISPR-based interventional data in human cells

Here we use recent data due to [23] involving large-scale perturbation experiments in human cells to further test performance. In the experiments used here, a CRISPR-based gene editing protocol was combined with single-cell RNA sequencing to allow for efficient multiplexed interventions. This protocol (described in detail in the reference) gives mRNA readouts for thousands of genes under each of a large number of interventions in two cell types, a leukemia cell line (K562) and retinal pigment epithelial (RPE) cells. For K562, the dataset includes $m = 2285$ detected genetic perturbations with a mean of 148 cells per perturbation and a median of 134 cells per perturbation. For RPE, $m = 2679$ genetic perturbations were detected with a mean of 101 cells per perturbation and a median of 79 cells per perturbation. After filtering, the causal graph of the K562 cell line contains $p = 8552$ nodes and the causal graph of the RPE cell line is of dimension $p = 8833$.

To ensure that models are tested on entirely unseen interventions, we split the data such that the set of interventions against which model output is tested is entirely disjoint from any model inputs.

To define causal status, we consider changes under intervention relative to the observational distribution. Treating the normalized gene expression levels as continuous random variables, we use an adapted version of the total variation between interventional and observational distributions:

$$V_{i,j} = \frac{1}{2} \int |f_j^{int(i)}(x) - f_j^{obs}(x)| dx$$

Specifically, if $V_{i,j} > \tau$ we consider the edge $i \rightarrow j$ causally relevant. In other words, if G^* is the gold-standard $p \times p$ adjacency matrix, we have $G_{ij}^* = 1 \iff V_{i,j} > \tau$. We chose $\tau = 0.3$ where the threshold τ is a problem specific hyperparameter. This approach focuses attention on strong causal effects and ensures that the observational distribution and the corresponding distribution under an intervention deviate sufficiently.

3.2 Baseline Comparisons

- **Simulated data:** We compare D²CL against: (i) IDA [3], (ii) SCL [18], and (iii) Marginal correlation coefficients.
- **Yeast Gene Deletion Experiments:** We compare D²CL against: (i) Classical causal structure learning approaches including IDA [3], LV-IDA [24], and the GIES algorithm [4]. (ii) A discriminative causal approach called SCL [18] and (iii) Marginal correlation coefficients (Pearson correlations in the figures, but Kendall and Spearman correlations gave similar results) as a simple baseline.
- **Human data:** We compare D²CL against: (i) IDA [3], (ii) GES [25] and (iii) GIES [4], (iv) LiNGAM [26], (v) CAM [27], and (vi) RFCI [28].

In addition to the established methods and baselines listed above we also considered a range of recently developed neural network-based causal learning approaches including SDI [29], DCDI [11], DCD-FG [12], ENCO [13] and CSIvA [14]. For scalability reasons, it is not feasible to apply SDI or DCDI to the datasets explored during the empirical evaluation (scalability issues for SDI and DCDI are also reported in [12]). At the time of writing, there was no source code available for CSIvA [14]. We evaluated DCD-FG and ENCO on simulated benchmark test cases with known ground truth graphs. Despite hyper-parameter tuning, we found that results were poor (see also Fig 3 in Main Text). This is likely due to the problem setting (including dataset type and sizes), in which $n = 6144$ observational and interventional training samples are available.

For applications motivating our work this is realistic, but nevertheless much smaller than the datasets used in the original papers (DCD-FG was applied to $p = 1000$ nodes using $n > 50000$ observational and interventional samples and ENCO to $p = 1000$ categorical variables using $n_{obs} = 10^5$ observational samples and $n_{int} = 4096$ interventional samples for each node). We emphasize that all of the above methods are conceptually exciting and very powerful in suitable settings. Our work is targeted at the high-dimensional, limited data regime – as relevant to many current, real-world applications, including in biology – hence our experimental set-up is designed to test performance in this kind of regime.

More generally, we note that the comparison with existing causal approaches is not one-to-one, since in many cases methods differ in their expected inputs and outputs. For example, IDA is aimed at analysis of observational data, hence the comparison is unfair since our approach has access also to background causal information Π . GIES allows for interventional data, but requires different inputs. Due to these differences in input/output requirements, we emphasize that comparisons here are provided for completeness but with the caveat that the various methods are intended for different use-cases (and furthermore make assumptions that are likely not met in the real biological data).

4 Additional Results

4.1 Gold-standard simulated benchmark data: Direct causal effects

Table S2: AUC values for direct cause-effect relations for $p = |V| = 1500$.

SNR	Linear				Pearson	MLP(tanh)				Pearson	MLP(leaky ReLU)				Pearson	Tanh				Pearson	Leaky ReLU				Pearson	Polynom 3			
	Pearson	IDA	D ² CL	SCL		IDA	D ² CL	SCL	Pearson		IDA	D ² CL	SCL	IDA		D ² CL	SCL	Pearson	IDA		D ² CL	SCL	IDA	D ² CL		SCL	Pearson	IDA	D ² CL
10.00	0.718	0.748	0.789	0.641	0.691	0.625	0.686	0.634	0.688	0.608	0.693	0.623	0.728	0.777	0.854	0.851	0.756	0.837	0.843	0.729	0.809	0.848	0.829	0.641					
6.00	0.700	0.771	0.795	0.617	0.670	0.609	0.658	0.638	0.666	0.590	0.683	0.602	0.710	0.783	0.861	0.841	0.736	0.818	0.839	0.692	0.784	0.824	0.821	0.637					
4.00	0.684	0.768	0.784	0.616	0.648	0.590	0.647	0.625	0.652	0.562	0.667	0.584	0.689	0.770	0.854	0.819	0.700	0.792	0.831	0.668	0.735	0.787	0.628	0.622					
2.00	0.638	0.781	0.802	0.615	0.613	0.544	0.639	0.594	0.617	0.521	0.661	0.592	0.639	0.750	0.844	0.764	0.644	0.743	0.815	0.630	0.651	0.722	0.777	0.617					
1.00	0.595	0.774	0.796	0.614	0.572	0.506	0.622	0.551	0.575	0.487	0.642	0.546	0.593	0.740	0.806	0.721	0.582	0.701	0.787	0.619	0.552	0.659	0.743	0.598					
0.75	0.589	0.765	0.793	0.612	0.556	0.494	0.619	0.566	0.567	0.483	0.638	0.568	0.580	0.724	0.795	0.689	0.572	0.696	0.783	0.610	0.539	0.645	0.734	0.603					
0.50	0.558	0.748	0.787	0.610	0.536	0.473	0.631	0.540	0.544	0.459	0.641	0.567	0.558	0.697	0.770	0.654	0.548	0.678	0.771	0.606	0.521	0.640	0.717	0.592					
0.25	0.537	0.735	0.784	0.572	0.530	0.467	0.617	0.517	0.496	0.434	0.624	0.552	0.538	0.667	0.733	0.588	0.517	0.667	0.748	0.579	0.514	0.634	0.694	0.543					
0.10	0.523	0.730	0.774	0.558	0.492	0.441	0.618	0.530	0.507	0.439	0.616	0.528	0.513	0.630	0.725	0.562	0.503	0.661	0.743	0.559	0.492	0.620	0.691	0.539					

Table S3: Problem Class I: AUPRC values for *direct* cause-effect relations for $p = |V| = 1500$.

SNR	Linear				Pearson	MLP(tanh)				Pearson	MLP(leaky ReLU)				Pearson	Tanh				Pearson	Leaky ReLU				Pearson	Polynom 3			
	Pearson	IDA	DDCL	SCL		IDA	DDCL	SCL	IDA		DDCL	SCL	IDA	DDCL		SCL	IDA	DDCL	SCL		IDA	DDCL	SCL	IDA		DDCL	SCL		
10.00	0.729	0.733	0.833	0.719	0.696	0.549	0.795	0.656	0.716	0.496	0.786	0.599	0.733	0.770	0.853	0.882	0.776	0.834	0.850	0.816	0.822	0.819	0.821	0.742					
6.00	0.698	0.745	0.754	0.711	0.663	0.521	0.784	0.631	0.683	0.493	0.783	0.530	0.707	0.766	0.879	0.860	0.756	0.796	0.844	0.782	0.792	0.771	0.821	0.723					
4.00	0.663	0.737	0.778	0.709	0.643	0.518	0.765	0.674	0.657	0.466	0.766	0.499	0.681	0.729	0.843	0.846	0.721	0.766	0.833	0.760	0.750	0.744	0.820	0.712					
2.00	0.605	0.731	0.806	0.700	0.589	0.484	0.739	0.567	0.613	0.445	0.784	0.621	0.622	0.676	0.848	0.829	0.661	0.704	0.799	0.709	0.644	0.630	0.771	0.668					
1.00	0.568	0.703	0.802	0.694	0.549	0.446	0.766	0.5	0.575	0.417	0.761	0.5	0.561	0.638	0.802	0.791	0.607	0.648	0.763	0.673	0.552	0.564	0.677	0.620					
0.75	0.571	0.698	0.798	0.694	0.537	0.441	0.719	0.634	0.558	0.421	0.767	0.674	0.539	0.603	0.801	0.755	0.583	0.634	0.764	0.665	0.541	0.544	0.675	0.609					
0.50	0.545	0.688	0.769	0.685	0.532	0.423	0.758	0.5	0.540	0.408	0.768	0.647	0.520	0.596	0.790	0.720	0.551	0.622	0.746	0.651	0.536	0.551	0.645	0.588					
0.25	0.521	0.676	0.784	0.5	0.517	0.417	0.754	0.5	0.479	0.394	0.763	0.594	0.509	0.592	0.758	0.644	0.508	0.599	0.726	0.643	0.518	0.548	0.617	0.5					
0.10	0.534	0.677	0.784	0.679	0.485	0.407	0.728	0.613	0.516	0.404	0.751	0.573	0.502	0.548	0.737	0.579	0.523	0.608	0.761	0.632	0.485	0.517	0.636	0.543					

4.2 Gold-standard simulated benchmark data: Total causal effects

Table S4: AUC values for indirect cause-effect relations for $p = |V| = 1500$.

SNR	Linear				Pearson	MLP(tanh)				Pearson	MLP(leaky ReLU)				Pearson	Tanh				Pearson	Leaky ReLU				Pearson	Polynom 3		
	Pearson	IDA	D ² CL	SCL		IDA	D ² CL	SCL	IDA		D ² CL	SCL	IDA	D ² CL		SCL	IDA	D ² CL	SCL		IDA	D ² CL	SCL					
10.00	0.553	0.907	0.928	0.708	0.548	0.522	0.733	0.700	0.563	0.483	0.789	0.738	0.511	0.903	0.947	0.905	0.502	0.857	0.943	0.839	0.610	0.822	0.933	0.761				
6.00	0.540	0.905	0.925	0.700	0.537	0.502	0.720	0.658	0.552	0.458	0.775	0.735	0.487	0.896	0.947	0.895	0.501	0.852	0.941	0.808	0.598	0.815	0.927	0.751				
4.00	0.530	0.905	0.928	0.677	0.533	0.490	0.711	0.675	0.548	0.447	0.767	0.727	0.460	0.881	0.947	0.888	0.504	0.848	0.937	0.782	0.581	0.803	0.914	0.732				
2.00	0.506	0.897	0.928	0.619	0.523	0.461	0.683	0.656	0.532	0.420	0.766	0.695	0.424	0.851	0.940	0.856	0.501	0.834	0.920	0.736	0.543	0.775	0.879	0.704				
1.00	0.507	0.888	0.925	0.609	0.513	0.433	0.660	0.626	0.518	0.393	0.738	0.672	0.400	0.810	0.895	0.791	0.502	0.824	0.900	0.676	0.520	0.753	0.831	0.658				
0.75	0.507	0.882	0.920	0.631	0.513	0.431	0.638	0.610	0.514	0.385	0.721	0.637	0.402	0.790	0.891	0.785	0.501	0.822	0.888	0.667	0.516	0.747	0.820	0.641				
0.50	0.506	0.877	0.918	0.505	0.510	0.422	0.618	0.577	0.513	0.387	0.713	0.655	0.407	0.771	0.861	0.740	0.502	0.821	0.880	0.651	0.510	0.742	0.808	0.635				
0.25	0.513	0.873	0.913	0.516	0.511	0.423	0.622	0.567	0.509	0.385	0.703	0.639	0.441	0.754	0.830	0.632	0.498	0.816	0.861	0.624	0.502	0.736	0.792	0.589				
0.10	0.506	0.867	0.905	0.542	0.501	0.417	0.617	0.571	0.503	0.385	0.705	0.613	0.481	0.741	0.819	0.595	0.500	0.817	0.865	0.598	0.503	0.734	0.784	0.537				

Table S5: Problem Class II: AUPRC values for *ancestral* cause-effect relations for $p = |V| = 1500$.

SNR	Linear				Pearson	MLP(tanh)				Pearson	MLP(leaky ReLU)				Pearson	Tanh				Pearson	Leaky ReLU				Pearson	Polynom 3			
	Pearson	IDA	DDCL	SCL		IDA	DDCL	SCL	IDA		DDCL	SCL	IDA	DDCL		SCL	IDA	DDCL	SCL		IDA	DDCL	SCL	IDA		DDCL	SCL		
10.00	0.581	0.878	0.892	0.632	0.545	0.547	0.750	0.680	0.545	0.490	0.727	0.809	0.594	0.899	0.926	0.883	0.526	0.829	0.916	0.816	0.639	0.794	0.904	0.690					
6.00	0.557	0.878	0.886	0.604	0.531	0.525	0.748	0.623	0.531	0.490	0.697	0.806	0.573	0.872	0.924	0.864	0.517	0.816	0.910	0.763	0.629	0.786	0.896	0.664					
4.00	0.543	0.879	0.890	0.581	0.523	0.521	0.731	0.624	0.525	0.479	0.714	0.785	0.554	0.872	0.922	0.854	0.513v	0.806	0.900	0.708	0.604	0.761	0.866	0.639					
2.00	0.512	0.853	0.878	0.519	0.517	0.484	0.701	0.635	0.513	0.457	0.700	0.790	0.522	0.818	0.905	0.795	0.505	0.790	0.834	0.640	0.565	0.715	0.816	0.601					
1.00	0.510	0.847	0.879	0.479	0.514	0.470	0.695	0.503	0.436	0.680	0.769	0.492	0.768	0.821	0.718	0.503	0.781	0.825	0.581	0.533	0.690	0.741	0.588						
0.75	0.517	0.839	0.861	0.554	0.517	0.483	0.669	0.636	0.502	0.442	0.676	0.759	0.483	0.747	0.822	0.700	0.504	0.788	0.811	0.590	0.523	0.692	0.732	0.577					
0.50	0.506	0.833	0.864	0.534	0.511	0.468	0.629	0.618	0.506	0.435	0.673	0.774	0.467	0.727	0.829	0.652	0.495	0.771	0.811	0.528	0.512	0.684	0.674	0.565					
0.25	0.523	0.831	0.870	0.572	0.518	0.467	0.577	0.522	0.506	0.436	0.649	0.712	0.479	0.714	0.751	0.593	0.500	0.776	0.814	0.537	0.506	0.669	0.694	0.565					
0.10	0.515	0.827	0.865	0.531	0.495	0.453	0.673	0.524	0.502	0.431	0.574	0.715	0.474	0.680	0.711	0.548	0.502	0.774	0.810	0.545	0.507	0.688	0.706	0.545					

4.4 Identifying Causal Directions

Table S7: Causal Directions: AUC values for test sets containing edges $A \rightarrow B$ and $B \rightarrow A$.

SNR	Linear	MLP(tanh)	MLP (leaky ReLU)	Tanh	Leaky ReLU	Polynom 3
10.00	0.983	0.711	0.787	1.000	0.994	0.929
6.00	0.981	0.659	0.806	1.000	0.990	0.920
4.00	0.989	0.648	0.802	1.000	0.989	0.927
2.00	0.988	0.721	0.777	1.000	0.992	0.929
1.00	0.987	0.668	0.798	0.999	0.990	0.924
0.75	0.997	0.618	0.765	0.998	0.993	0.926
0.50	0.990	0.669	0.784	0.994	0.991	0.920
0.25	0.991	0.683	0.721	0.962	0.991	0.905
0.10	0.985	0.674	0.790	0.941	0.992	0.915

4.5 Multiplicative Noise

Table S8: Gold-standard simulations: AUROC values for *direct* cause-effect relations for $p = |V| = 1500$: additive and multiplicative noise

SNR	deterministic hard interventions				stochastic hard interventions			
	additive Noise		multiplicative Noise		additive Noise		multiplicative Noise	
	Linear	Tanh	Linear	Tanh	Linear	Tanh	Linear	Tanh
10.00	0.849	0.882	0.746	0.810	0.794	0.812	0.696	0.796
6.00	0.799	0.891	0.691	0.779	0.781	0.826	0.688	0.790
4.00	0.805	0.875	0.675	0.771	0.785	0.825	0.677	0.791
2.00	0.833	0.876	0.664	0.775	0.779	0.814	0.676	0.778
1.00	0.833	0.837	0.678	0.768	0.771	0.734	0.675	0.748
0.75	0.828	0.832	0.666	0.747	0.770	0.718	0.669	0.749
0.50	0.818	0.798	0.643	0.757	0.737	0.726	0.672	0.771
0.25	0.816	0.772	0.639	0.745	0.720	0.709	0.673	0.743
0.10	0.808	0.761	0.617	0.753	0.720	0.697	0.663	0.751

Table S9: Gold-standard simulations: AUPRC values for *direct* cause-effect relations for $p = |V| = 1500$: additive and multiplicative noise

SNR	deterministic hard interventions				stochastic hard interventions			
	additive Noise		multiplicative Noise		additive Noise		multiplicative Noise	
	Linear	Tanh	Linear	Tanh	Linear	Tanh	Linear	Tanh
10.00	0.834	0.853	0.691	0.776	0.768	0.795	0.692	0.784
6.00	0.755	0.879	0.645	0.745	0.767	0.787	0.655	0.757
4.00	0.779	0.843	0.631	0.739	0.758	0.791	0.639	0.765
2.00	0.807	0.848	0.609	0.743	0.753	0.780	0.621	0.742
1.00	0.803	0.803	0.626	0.732	0.753	0.689	0.615	0.704
0.75	0.798	0.801	0.627	0.714	0.734	0.668	0.634	0.699
0.50	0.770	0.791	0.617	0.717	0.748	0.680	0.625	0.735
0.25	0.785	0.758	0.611	0.718	0.732	0.676	0.628	0.705
0.10	0.784	0.737	0.599	0.718	0.739	0.664	0.607	0.715

4.6 Out-of-distribution tests

Table S10: Out-of-distribution experiments: Cross validation experiments - AUPRC values

			Predicted class																	
			Linear										LeakyReLU							
			SNR	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.10	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25
Trained class	Linear	10.00	0.757	0.762	0.763	0.765	0.758	0.752	0.743	0.726	0.703	0.715	0.719	0.718	0.708	0.680	0.675	0.666	0.651	0.647
		6.00	0.767	0.769	0.768	0.767	0.752	0.746	0.736	0.718	0.691	0.713	0.713	0.709	0.691	0.666	0.661	0.656	0.648	0.647
		4.00	0.755	0.761	0.764	0.769	0.759	0.752	0.742	0.722	0.699	0.678	0.683	0.683	0.679	0.664	0.663	0.658	0.653	0.647
		2.00	0.742	0.749	0.757	0.770	0.765	0.760	0.748	0.727	0.692	0.669	0.674	0.677	0.688	0.679	0.677	0.668	0.656	0.649
		1.00	0.707	0.714	0.723	0.744	0.753	0.755	0.753	0.739	0.704	0.652	0.658	0.660	0.679	0.679	0.676	0.670	0.658	0.650
		0.75	0.685	0.695	0.706	0.736	0.753	0.753	0.751	0.738	0.718	0.585	0.593	0.605	0.643	0.650	0.654	0.659	0.660	0.650
		0.50	0.679	0.687	0.697	0.729	0.746	0.750	0.753	0.745	0.730	0.606	0.609	0.617	0.636	0.651	0.656	0.656	0.658	0.655
	Tanh	0.25	0.646	0.650	0.658	0.693	0.721	0.730	0.740	0.741	0.733	0.576	0.579	0.584	0.605	0.631	0.637	0.641	0.650	0.656
		0.10	0.629	0.633	0.642	0.670	0.692	0.700	0.711	0.725	0.736	0.614	0.617	0.621	0.630	0.626	0.625	0.623	0.629	0.649
		10.00	0.685	0.689	0.690	0.689	0.667	0.656	0.642	0.625	0.603	0.765	0.764	0.753	0.720	0.675	0.667	0.655	0.622	0.610
		6.00	0.683	0.686	0.687	0.685	0.663	0.652	0.638	0.618	0.595	0.759	0.761	0.754	0.717	0.670	0.662	0.651	0.620	0.606
		4.00	0.683	0.687	0.688	0.688	0.664	0.652	0.640	0.623	0.599	0.751	0.752	0.746	0.713	0.661	0.658	0.646	0.616	0.607
		2.00	0.674	0.677	0.677	0.687	0.667	0.652	0.633	0.615	0.599	0.707	0.709	0.713	0.712	0.651	0.648	0.633	0.605	0.596
		1.00	0.686	0.688	0.688	0.699	0.701	0.691	0.672	0.652	0.640	0.684	0.690	0.691	0.700	0.695	0.688	0.666	0.641	0.633
	0.75	0.675	0.679	0.681	0.694	0.703	0.698	0.687	0.658	0.647	0.675	0.682	0.683	0.697	0.699	0.691	0.670	0.651	0.632	
	0.50	0.681	0.685	0.688	0.700	0.703	0.699	0.685	0.663	0.650	0.638	0.645	0.656	0.674	0.687	0.686	0.673	0.652	0.631	
	0.25	0.645	0.649	0.654	0.672	0.696	0.700	0.703	0.688	0.667	0.549	0.551	0.564	0.597	0.617	0.626	0.645	0.659	0.650	
	0.10	0.630	0.633	0.633	0.641	0.658	0.670	0.683	0.696	0.689	0.483	0.485	0.481	0.483	0.514	0.543	0.585	0.638	0.661	

			Predicted class																	
			Polynom 3										Tanh							
			SNR	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.1	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25
Trained class	Linear	10.00	0.678	0.675	0.675	0.666	0.645	0.643	0.633	0.626	0.616	0.707	0.708	0.708	0.704	0.677	0.667	0.645	0.617	0.595
		6.00	0.701	0.698	0.694	0.674	0.650	0.644	0.632	0.623	0.619	0.723	0.722	0.721	0.709	0.677	0.665	0.644	0.619	0.598
		4.00	0.656	0.654	0.656	0.657	0.641	0.640	0.632	0.627	0.619	0.690	0.691	0.694	0.695	0.672	0.665	0.643	0.614	0.598
		2.00	0.650	0.648	0.649	0.652	0.646	0.645	0.636	0.629	0.621	0.693	0.695	0.697	0.698	0.678	0.669	0.648	0.620	0.597
		1.00	0.633	0.631	0.632	0.635	0.631	0.631	0.627	0.627	0.623	0.671	0.672	0.673	0.672	0.660	0.656	0.644	0.623	0.600
		0.75	0.549	0.554	0.571	0.606	0.622	0.626	0.626	0.631	0.625	0.626	0.631	0.638	0.653	0.651	0.650	0.643	0.626	0.599
		0.50	0.583	0.590	0.608	0.627	0.626	0.628	0.625	0.630	0.627	0.653	0.657	0.661	0.669	0.659	0.655	0.643	0.623	0.600
	Tanh	0.25	0.530	0.530	0.537	0.570	0.598	0.608	0.612	0.622	0.627	0.592	0.596	0.604	0.620	0.633	0.636	0.630	0.620	0.603
		0.10	0.571	0.572	0.577	0.598	0.604	0.606	0.603	0.608	0.620	0.627	0.629	0.634	0.642	0.638	0.637	0.626	0.612	0.601
		10.00	0.752	0.746	0.737	0.701	0.656	0.637	0.619	0.590	0.586	0.806	0.803	0.797	0.757	0.684	0.653	0.610	0.590	0.575
		6.00	0.752	0.750	0.744	0.709	0.662	0.640	0.620	0.591	0.585	0.805	0.808	0.805	0.765	0.682	0.649	0.608	0.591	0.574
		4.00	0.735	0.737	0.736	0.707	0.662	0.637	0.617	0.588	0.584	0.798	0.803	0.808	0.778	0.690	0.657	0.607	0.586	0.573
		2.00	0.695	0.696	0.694	0.695	0.662	0.634	0.610	0.578	0.571	0.749	0.757	0.771	0.794	0.720	0.690	0.629	0.580	0.560
		1.00	0.667	0.668	0.671	0.676	0.673	0.656	0.627	0.606	0.599	0.718	0.720	0.727	0.744	0.735	0.720	0.661	0.598	0.587
	0.75	0.660	0.660	0.662	0.669	0.663	0.660	0.640	0.616	0.595	0.702	0.705	0.710	0.718	0.706	0.699	0.667	0.609	0.577	
	0.50	0.620	0.621	0.625	0.644	0.651	0.653	0.641	0.621	0.600	0.653	0.656	0.663	0.673	0.678	0.674	0.656	0.612	0.576	
	0.25	0.539	0.543	0.543	0.560	0.586	0.591	0.616	0.627	0.620	0.562	0.565	0.568	0.582	0.593	0.596	0.619	0.634	0.596	
	0.10	0.520	0.520	0.518	0.523	0.526	0.537	0.555	0.610	0.628	0.503	0.505	0.507	0.508	0.512	0.517	0.540	0.598	0.617	

			Predicted class																	
			MLP (leaky ReLU)										MLP (Tanh)							
			SNR	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.1	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25
Trained class	Linear	10.00	0.459	0.452	0.447	0.427	0.433	0.435	0.437	0.441	0.447	0.466	0.455	0.447	0.434	0.442	0.444	0.446	0.455	0.459
		6.00	0.454	0.451	0.450	0.431	0.435	0.437	0.438	0.443	0.448	0.459	0.454	0.449	0.436	0.443	0.446	0.447	0.456	0.459
		4.00	0.492	0.487	0.480	0.457	0.448	0.445	0.444	0.444	0.449	0.494	0.487	0.477	0.461	0.450	0.451	0.448	0.453	0.461
		2.00	0.488	0.481	0.467	0.440	0.438	0.440	0.441	0.444	0.448	0.490	0.483	0.469	0.450	0.446	0.445	0.444	0.456	0.457
		1.00	0.503	0.500	0.495	0.464	0.447	0.445	0.443	0.444	0.449	0.497	0.493	0.489	0.471	0.455	0.450	0.446	0.453	0.458
		0.75	0.514	0.510	0.501	0.457	0.443	0.442	0.440	0.442	0.445	0.520	0.514	0.504	0.468	0.451	0.449	0.446	0.452	0.456
		0.50	0.489	0.485	0.474	0.446	0.439	0.440	0.439	0.441	0.446	0.496	0.487	0.478	0.457	0.449	0.447	0.446	0.454	0.457
	Tanh	0.25	0.496	0.495	0.492	0.466	0.452	0.449	0.443	0.444	0.444	0.504	0.500	0.497	0.479	0.461	0.455	0.450	0.453	0.456
		0.10	0.468	0.467	0.465	0.453	0.447	0.444	0.441	0.444	0.443	0.477	0.472	0.469	0.458	0.449	0.453	0.450	0.453	0.457
		10.00	0.450	0.443	0.431	0.427	0.438	0.442	0.448	0.457	0.464	0.457	0.450	0.436	0.437	0.450	0.452	0.456	0.467	0.477
		6.00	0.436	0.434	0.432	0.432	0.441	0.444	0.449	0.458	0.458	0.444	0.439	0.438	0.440	0.448	0.452	0.458	0.468	0.473
		4.00	0.448	0.443	0.435	0.429	0.437	0.440	0.446	0.455	0.461	0.451	0.443	0.438	0.435	0.446	0.450	0.452	0.468	0.476
		2.00	0.450	0.449	0.448	0.436	0.437	0.441	0.444	0.455	0.464	0.451	0.452	0.454	0.444	0.445	0.448	0.450	0.467	0.478
		1.00	0.447	0.442	0.441	0.440	0.438	0.443	0.447	0.450	0.454	0.447	0.444	0.443	0.442	0.447	0.447	0.445	0.459	0.466
	0.75	0.474	0.474	0.470	0.450	0.451	0.451	0.445	0.450	0.452	0.477	0.479	0.475	0.459	0.458	0.455	0.453	0.457	0.462	
	0.50	0.486	0.493	0.497	0.500	0.485	0.467	0.456	0.450	0.451	0.490	0.493	0.500	0.506	0.481	0.467	0.456	0.453	0.457	
	0.25	0.524	0.528	0.529	0.536	0.519	0.514	0.503	0.477	0.459	0.516	0.517	0.519	0.523	0.514	0.509	0.507	0.480	0.460	
	0.10	0.519	0.525	0.524	0.521	0.529	0.540	0.556	0.572	0.500	0.517	0.519	0.520	0.518	0.534	0.532	0.555	0.580	0.485	

Table S11: Out-of-distribution experiments: Cross validation experiments - AUROC values

			Predicted class																	
			Linear										leaky ReLU							
			10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.1	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.10
Trained class	Linear	10.00	0.787	0.791	0.793	0.793	0.786	0.781	0.772	0.755	0.728	0.752	0.756	0.756	0.746	0.730	0.725	0.718	0.705	0.701
		6.00	0.796	0.798	0.799	0.797	0.783	0.778	0.768	0.749	0.722	0.752	0.754	0.753	0.743	0.726	0.720	0.714	0.708	0.705
		4.00	0.792	0.797	0.800	0.803	0.791	0.784	0.774	0.756	0.730	0.725	0.731	0.735	0.738	0.730	0.726	0.719	0.710	0.703
		2.00	0.763	0.770	0.781	0.797	0.793	0.789	0.780	0.760	0.728	0.708	0.715	0.721	0.733	0.730	0.727	0.721	0.713	0.706
		1.00	0.732	0.738	0.752	0.781	0.791	0.792	0.790	0.776	0.746	0.692	0.699	0.704	0.724	0.732	0.730	0.728	0.718	0.709
		0.75	0.710	0.716	0.732	0.768	0.785	0.786	0.786	0.776	0.755	0.652	0.661	0.672	0.702	0.712	0.714	0.720	0.722	0.709
	Tanh	0.50	0.709	0.715	0.727	0.763	0.782	0.785	0.788	0.782	0.769	0.670	0.676	0.683	0.700	0.710	0.715	0.718	0.721	0.716
		0.25	0.673	0.677	0.689	0.731	0.760	0.768	0.776	0.778	0.774	0.628	0.632	0.638	0.659	0.688	0.696	0.704	0.714	0.721
		0.10	0.662	0.667	0.678	0.712	0.736	0.742	0.751	0.763	0.776	0.674	0.678	0.682	0.687	0.688	0.687	0.687	0.694	0.714
		10.00	0.654	0.656	0.656	0.656	0.644	0.636	0.627	0.619	0.608	0.788	0.786	0.778	0.747	0.697	0.685	0.666	0.632	0.619
		6.00	0.653	0.654	0.654	0.652	0.636	0.628	0.619	0.610	0.598	0.781	0.781	0.775	0.743	0.696	0.684	0.666	0.630	0.610
		4.00	0.649	0.651	0.651	0.651	0.639	0.630	0.622	0.614	0.604	0.777	0.777	0.773	0.744	0.691	0.678	0.659	0.622	0.611
		2.00	0.642	0.644	0.644	0.648	0.635	0.624	0.613	0.604	0.600	0.732	0.734	0.736	0.739	0.684	0.670	0.646	0.611	0.600
		1.00	0.672	0.673	0.674	0.682	0.681	0.673	0.659	0.645	0.637	0.711	0.716	0.721	0.736	0.739	0.726	0.705	0.673	0.650
0.75	0.664	0.667	0.669	0.682	0.691	0.687	0.676	0.656	0.644	0.708	0.715	0.720	0.737	0.746	0.737	0.718	0.694	0.664		
0.50	0.673	0.676	0.678	0.689	0.694	0.690	0.677	0.657	0.644	0.683	0.692	0.703	0.723	0.732	0.733	0.720	0.697	0.663		
0.25	0.679	0.682	0.685	0.696	0.712	0.711	0.707	0.687	0.663	0.583	0.592	0.607	0.647	0.676	0.687	0.708	0.720	0.698		
0.10	0.674	0.676	0.678	0.686	0.702	0.708	0.712	0.713	0.694	0.501	0.504	0.502	0.505	0.549	0.588	0.638	0.697	0.718		

			Predicted class																		
			Polynom 3										Tanh								
			10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.1	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.10	
Trained class	Linear	SNR	10.00	0.728	0.724	0.721	0.707	0.687	0.686	0.675	0.663	0.656	0.758	0.757	0.756	0.748	0.725	0.716	0.695	0.664	0.639
		6.00	0.739	0.735	0.729	0.712	0.689	0.685	0.674	0.667	0.665	0.768	0.766	0.764	0.752	0.723	0.712	0.692	0.669	0.648	
		4.00	0.709	0.708	0.711	0.707	0.691	0.689	0.680	0.669	0.663	0.749	0.750	0.750	0.748	0.728	0.719	0.698	0.667	0.645	
		2.00	0.690	0.686	0.687	0.692	0.687	0.688	0.680	0.673	0.668	0.742	0.742	0.743	0.742	0.723	0.717	0.697	0.672	0.648	
		1.00	0.676	0.673	0.673	0.673	0.673	0.675	0.678	0.678	0.675	0.670	0.718	0.718	0.719	0.718	0.709	0.706	0.695	0.678	0.650
		0.75	0.607	0.611	0.625	0.655	0.668	0.672	0.674	0.677	0.671	0.694	0.696	0.700	0.705	0.699	0.698	0.690	0.682	0.653	
	Tanh	0.50	0.635	0.639	0.654	0.670	0.670	0.671	0.671	0.675	0.673	0.710	0.712	0.714	0.714	0.702	0.699	0.688	0.677	0.656	
		0.25	0.563	0.562	0.569	0.608	0.645	0.655	0.661	0.668	0.674	0.644	0.648	0.656	0.670	0.679	0.680	0.675	0.671	0.660	
		0.10	0.609	0.608	0.613	0.637	0.651	0.654	0.653	0.656	0.666	0.680	0.681	0.686	0.691	0.685	0.682	0.670	0.656	0.652	
		10.00	0.796	0.792	0.785	0.742	0.683	0.663	0.641	0.608	0.599	0.841	0.840	0.834	0.792	0.715	0.683	0.640	0.605	0.588	
		6.00	0.792	0.791	0.785	0.750	0.694	0.672	0.650	0.612	0.592	0.840	0.841	0.839	0.801	0.720	0.686	0.644	0.609	0.582	
		4.00	0.780	0.780	0.779	0.749	0.691	0.666	0.644	0.604	0.591	0.835	0.840	0.843	0.812	0.726	0.693	0.643	0.600	0.579	
		2.00	0.738	0.739	0.740	0.749	0.702	0.672	0.641	0.596	0.581	0.797	0.805	0.815	0.832	0.754	0.721	0.659	0.596	0.569	
		1.00	0.705	0.707	0.709	0.721	0.720	0.704	0.673	0.642	0.625	0.761	0.766	0.775	0.792	0.779	0.765	0.704	0.638	0.605	
0.75	0.699	0.702	0.703	0.715	0.714	0.710	0.690	0.655	0.633	0.750	0.753	0.758	0.768	0.762	0.758	0.721	0.660	0.611			
0.50	0.667	0.667	0.673	0.687	0.693	0.698	0.690	0.659	0.633	0.700	0.703	0.709	0.719	0.723	0.724	0.711	0.661	0.610			
0.25	0.555	0.561	0.564	0.587	0.625	0.638	0.662	0.681	0.660	0.591	0.596	0.603	0.621	0.639	0.644	0.666	0.688	0.643			
0.10	0.545	0.547	0.544	0.553	0.552	0.566	0.590	0.649	0.678	0.524	0.528	0.530	0.528	0.529	0.538	0.568	0.637	0.673			

			Predicted class																		
			MLP (leaky ReLU)										MLP (Tanh)								
			10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.1	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.10	
Trained class	Linear	SNR	10.00	0.400	0.392	0.382	0.362	0.376	0.383	0.388	0.396	0.403	0.413	0.402	0.392	0.378	0.391	0.394	0.395	0.404	0.416
		6.00	0.394	0.386	0.384	0.368	0.378	0.384	0.388	0.397	0.405	0.407	0.397	0.393	0.381	0.392	0.396	0.396	0.407	0.418	
		4.00	0.443	0.440	0.434	0.404	0.395	0.395	0.393	0.396	0.403	0.450	0.446	0.443	0.417	0.405	0.403	0.397	0.403	0.417	
		2.00	0.434	0.430	0.419	0.388	0.388	0.390	0.390	0.398	0.404	0.437	0.434	0.424	0.403	0.401	0.397	0.395	0.408	0.415	
		1.00	0.453	0.451	0.450	0.429	0.403	0.400	0.393	0.398	0.401	0.451	0.449	0.448	0.435	0.414	0.403	0.397	0.404	0.409	
		0.75	0.467	0.465	0.460	0.423	0.398	0.394	0.391	0.396	0.398	0.469	0.466	0.463	0.435	0.408	0.401	0.396	0.403	0.407	
	Tanh	0.50	0.447	0.444	0.435	0.407	0.392	0.392	0.389	0.396	0.399	0.452	0.448	0.442	0.419	0.404	0.399	0.396	0.403	0.408	
		0.25	0.455	0.454	0.453	0.432	0.414	0.409	0.398	0.398	0.397	0.464	0.463	0.461	0.448	0.425	0.415	0.405	0.404	0.406	
		0.10	0.429	0.425	0.420	0.407	0.401	0.397	0.394	0.400	0.398	0.440	0.434	0.429	0.418	0.408	0.410	0.403	0.403	0.408	
		10.00	0.398	0.386	0.368	0.358	0.383	0.397	0.413	0.429	0.444	0.417	0.404	0.388	0.381	0.406	0.418	0.426	0.439	0.456	
		6.00	0.379	0.371	0.364	0.362	0.382	0.395	0.412	0.432	0.425	0.400	0.390	0.385	0.381	0.400	0.412	0.424	0.441	0.442	
		4.00	0.391	0.384	0.374	0.358	0.378	0.391	0.405	0.430	0.439	0.408	0.399	0.388	0.374	0.397	0.407	0.415	0.440	0.455	
		2.00	0.400	0.399	0.401	0.382	0.379	0.390	0.395	0.427	0.451	0.407	0.409	0.416	0.400	0.395	0.401	0.406	0.441	0.464	
		1.00	0.399	0.393	0.391	0.391	0.386	0.395	0.397	0.405	0.418	0.403	0.399	0.400	0.399	0.403	0.401	0.394	0.414	0.432	
0.75	0.442	0.435	0.431	0.403	0.405	0.404	0.398	0.402	0.409	0.446	0.445	0.439	0.419	0.415	0.409	0.403	0.409	0.419			
0.50	0.463	0.469	0.472	0.480	0.460	0.436	0.414	0.406	0.402	0.468	0.471	0.478	0.487	0.460	0.433	0.413	0.408	0.410			
0.25	0.503	0.504	0.506	0.515	0.495	0.487	0.473	0.446	0.412	0.493	0.496	0.497	0.503	0.490	0.482	0.478	0.444	0.413			
0.10	0.501	0.503	0.501	0.501	0.508	0.519	0.538	0.547	0.477	0.497	0.497	0.498	0.499	0.518	0.516	0.537	0.559	0.460			

4.7 Yeast Gene Deletion Experiments

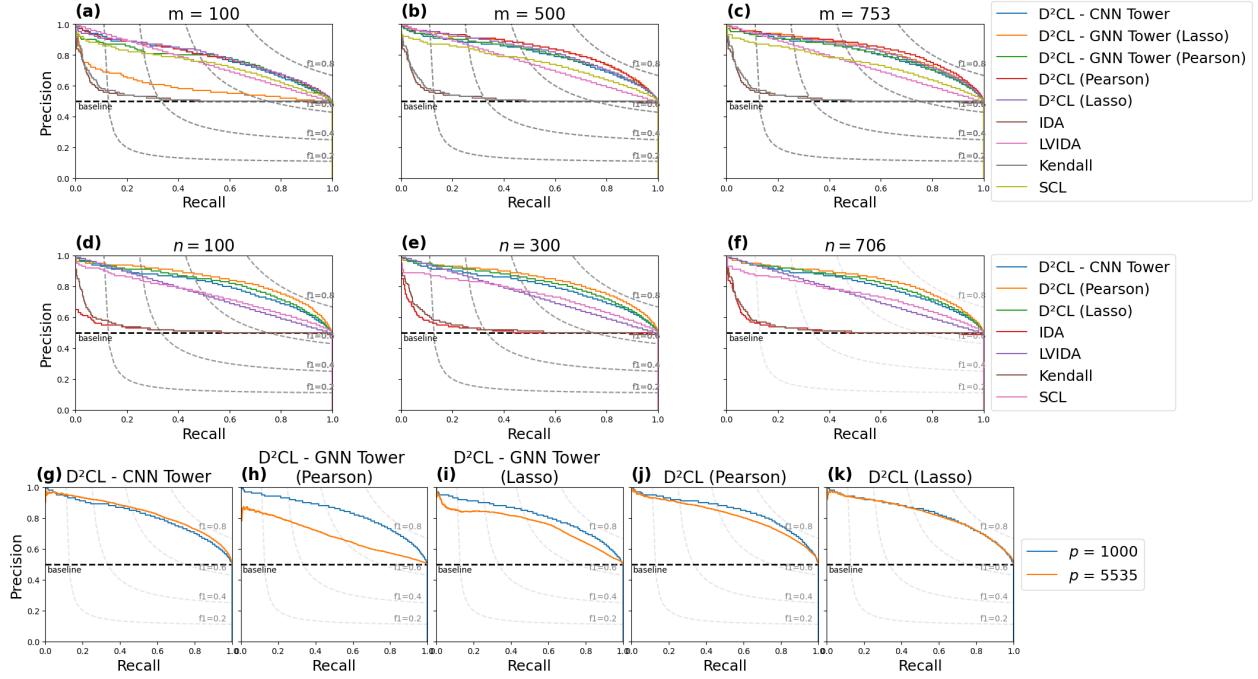


Figure S2: Results - yeast gene deletion experiments. Causal learning methods, including D²CL, were applied to gene expression measurements from yeast cells. Performance was quantified using causal Precision-Recall curves (and the area under the PR-curves, or AUPRC see Table S12) computed with respect to a causal ground truth obtained from entirely unseen interventional experiments (see Text for details). Panels (a)–(c): the number of interventions whose effects are available to the learner is varied as shown (with problem dimension fixed to $p=1000$ and sample size to $n=706$). Panels (d)–(f): the sample size n of the data matrix X is varied as shown (with problem dimension fixed to $p=1000$ and number of available interventions fixed to $m=753$). Panels (g)–(k): analogous results for a higher-dimensional setting covering all available genes (roughly the full yeast genome) with $p=5535$ (with $n=706$ and $m=753$). Here, only D²CL variants are shown, as the other methods could not be run due to the computational burden in this higher dimensional case. Comparison with the corresponding $p=1000$ case demonstrates the scalability of D²CL, with performance broadly maintained in the higher dimensional setting. [D²CL variants shown include a CNN tower alone, GNN tower alone and the entire D²CL architecture; methods compared against include IDA, LVIDA, Kendall correlations (as a non-causal baseline) and SCL (see text and SI for details and references). For D²CL and its variants two different initial graph estimates were used based respectively on Pearson correlation coefficients (“Pearson”) and on a lightweight regression (“Lasso”; see Text for details).]

Table S12: Yeast Gene Deletion - AUPRC scores: Causal learning methods, including D²CL, were applied to gene expression measurements from yeast cells. Performance was quantified using causal area under Precision-Recall curves. Subtables extend results of Figure S2, Subtable S12a extends panels (a)-(c), Subtable S12b provides AUPRC scores for panels (d)-(f), and Subtable S12c presents scores for panels (g)-(k).

(a) Varying amount of causal input m				(b) Varying sample size n			
	m=100	m=500	m=753		n=100	n=300	n=706
D ² CL - CNN Tower	0.784	0.804	0.815	D ² CL - CNN Tower	0.806	0.807	0.815
D ² CL - GNN Tower (Lasso)	0.597	0.837	0.835	D ² CL (Pearson)	0.846	0.845	0.845
D ² CL - GNN Tower (Pearson)	0.766	0.811	0.814	D ² CL (Lasso)	0.824	0.827	0.828
D ² CL (Pearson)	0.787	0.839	0.845	IDA	0.524	0.533	0.533
D ² CL (Lasso)	0.792	0.822	0.828	LVIDA	0.499	0.499	0.499
IDA	0.533	0.533	0.533	Kendall	0.536	0.543	0.538
LVIDA	0.499	0.499	0.499	SCL	0.749	0.749	0.749
Kendall	0.538	0.538	0.538				
SCL	0.737	0.752	0.749				
(c) Scaling to full yeast genome							
	p=1000		p=5535				
D ² CL - CNN Tower	0.815		0.829				
D ² CL - GNN Tower (Pearson)	0.835		0.683				
D ² CL - GNN Tower (Lasso)	0.814		0.753				
D ² CL (Pearson)	0.845		0.811				
D ² CL (Lasso)	0.828		0.821				

4.8 Wall-clock times

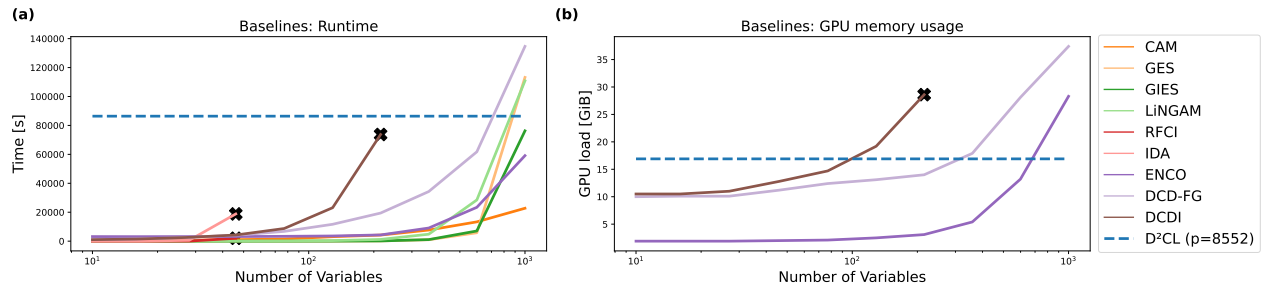


Figure S3: Wall-clock time and GPU memory usage: (a) Runtime performance of baseline algorithms as the number of nodes increases ($n = (10, \dots, 1000)$). Black crosses mark experiments terminated due to runtime constraints. (b) GPU memory requirements as the number of nodes increases for ENCO, DCDI, and DCD-FG. D²CL's wall-clock time and GPU memory consumption for a graph with $n = 8852$ nodes are indicated by the dashed blue horizontal line.

5 Data Generation: Pseudo-Code

List of Variables/Abbreviations

$\mathcal{D}_{input}^k$	Input parent data for node k , $size = [N_{samples} \times N_{parents}]$
$\mathcal{F}$	Edge transition function, either linear, nonlinear, or MLP; $\mathcal{F} : \mathbb{R}^{N_{samples} \times N_{parents}} \rightarrow \mathbb{R}^{N_{samples} \times 1}$
$\mathcal{N}(\mu, \sigma)$	Gaussian distribution with mean μ and standard deviation σ
σ	Nonlinear activation function in MLP, $\sigma \in \{tanh, leakyReLU, \}$
$\tau_{low}^{init}, \tau_{high}^{init}$	Lower and upper thresholds for initialization of root node distributions
$argwhere(condition)$	Vector function returning all indices where the condition is true
b	Bias vector
G	Directed acyclic graph
g	Nonlinear edge transition function, $g \in \{tanh, leakyReLU, Polynom3\}$
$leakyReLU$	Leaky Rectifier Linear Unit
$N_{parents}$	Number of parents for node k
N_{roots}	Number of root nodes in DAG G
$N_{samples}$	Number of samples observed for node k
$parents$	Set of parent nodes for node k
$Polynom3$	Polynomial function with maximum degree 3
SNR_{exo}	Signal-to-noise ratio for (node dependent) exogeneous noise
SNR_{meas}	Signal-to-noise ratio for measurement noise
$tanh$	Tangent hyperbolic
$Uniform(\tau_{low}, \tau_{high})$	Uniform distribution between lower value τ_{low} and upper value τ_{high}
W, W_1, W_2	Weight matrices

5.1 SEM Data Generation

Algorithm 1 SEM data generation

```

1: INPUT:  $G, \mathcal{F}, SNR_{exo}, SNR_{meas}, \tau_{low}^{init}, \tau_{high}^{init}$ 
2: RETURN: data
3:
4: Extract layers from  $G$ 
5: Init result:  $data = zeros(N_{samples}, N_{variables})$ 
6: for  $iteration = 0, 1, \dots, N_{roots} - 1$  do
7:   Sample mean and std from  $Uniform(\tau_{low}^{init}, \tau_{high}^{init})$ 
8:   Sample obs from Gaussian:  $data[i, :] \sim \mathcal{N}(mean, std)$ 
9: end for
10:
11: for  $layer$  in  $layers[1 \dots end]$  do
12:   for  $node$  in  $layer$  do
13:     # Get parent nodes:
14:      $parents = argwhere(G.adjacency[:, node] == True)$ 
15:
16:     # Compute node state:
17:      $nodeState = \mathcal{F}(parents)$ 
18:
19:     # Add exogenous noise  $size = [N_{samples}, 1]$ :
20:      $nodeState += \mathcal{N}(0, \frac{|mean(nodeState)|}{SNR_{exo}})$ 
21:   end for
22: end for
23:
24: # Add measurement noise  $size = [N_{samples}, N_{variables}]$ :
25:  $data += \mathcal{N}(0, \frac{|mean(data, axis=0)|}{SNR_{meas}})$ 

```

5.2 Functions

1. Linear:

Hyperparameters:

(a) $W = Uniform([W_{low}, W_{high}])$

(b) $b = Uniform([b_{low}, b_{high}])$

Equation: $\mathcal{F} = W\mathcal{D}_{input} + b$

2. Nonlinear:

hyperparameters:

(a) nonlinear function $g \in \{tanh, leakyReLU, Polynom3\}$

Equation: $\mathcal{F} = g(sum(\mathcal{D}_{input}, axis = 1))$

3. MLP:

Hyperparameters:

(a) activation function $\sigma \in \{tanh, leakyReLU, \}$

(b) $W_1 = Uniform([W_{low}, W_{high}])$

(c) $W_2 = Uniform([W_{low}, W_{high}])$

(d) $b = Uniform([b_{low}, b_{high}])$

Equation: $\mathcal{F} = W_2(\sigma(norm(W_1\mathcal{D}_{input} + b)))$

References

- [1] Spirtes, P., Glymour, C. & Scheines, R. *Causation, Prediction, and Search* (MIT Press, Cambridge, 2000).
- [2] Shimizu, S., Hoyer, P. O., Hyvärinen, A. & Kerminen, A. A linear non-Gaussian acyclic model for causal discovery. *The Journal of Machine Learning Research* **7**, 2003–2030 (2006).
- [3] Maathuis, M. H., Kalisch, M. & Bühlmann, P. Estimating high-dimensional intervention effects from observational data. *The Annals of Statistics* **37**, 3133–3164 (2009).
- [4] Hauser, A. & Bühlmann, P. Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research* **13**, 2409–2464 (2012).
- [5] Colombo, D., Maathuis, M. H., Kalisch, M. & Richardson, T. S. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics* **40**, 294–321 (2012).
- [6] Peters, J., Bühlmann, P. & Meinshausen, N. Causal inference using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society* **78**, 947–1012 (2016).
- [7] Hill, S. M., Oates, C. J., Blythe, D. A. & Mukherjee, S. Causal learning via manifold regularization. *The Journal of Machine Learning Research* **20**, 1–32 (2019).
- [8] Eigenmann, M., Mukherjee, S. & Maathuis, M. Evaluation of causal structure learning algorithms via risk estimation. In *Proceedings of Uncertainty in Artificial Intelligence 2020, UAI 2020* (2020).
- [9] Zheng, X., Aragam, B., Ravikumar, P. K. & Xing, E. P. Dags with no tears: Continuous optimization for structure learning. *Advances in neural information processing systems* **31** (2018).
- [10] Ke, N. R. *et al.* Learning neural causal models from unknown interventions. *arXiv preprint arXiv:1910.01075* (2019).
- [11] Brouillard, P., Lachapelle, S., Lacoste, A., Lacoste-Julien, S. & Drouin, A. Differentiable causal discovery from interventional data. *Advances in Neural Information Processing Systems* **33**, 21865–21877 (2020).
- [12] Lopez, R., Hütter, J.-C., Pritchard, J. & Regev, A. Large-scale differentiable causal discovery of factor graphs. *Advances in Neural Information Processing Systems* **35**, 19290–19303 (2022).
- [13] Lippe, P., Cohen, T. & Gavves, E. Efficient neural causal discovery without acyclicity constraints. *arXiv preprint arXiv:2107.10483* (2021).
- [14] Ke, N. R. *et al.* Learning to induce causal structure. *arXiv preprint arXiv:2204.04875* (2022).
- [15] Kemmeren, P. *et al.* Large-scale genetic perturbations reveal regulatory networks and an abundance of gene-specific repressors. *Cell* **157**, 740–752 (2014).
- [16] Meinshausen, N. *et al.* Methods for causal inference from gene perturbation experiments and validation. *Proceedings of the National Academy of Sciences of the United States of America* **113**, 7361–7368 (2016).
- [17] Zhang, J. Causal reasoning with ancestral graphs. *The Journal of Machine Learning Research* **9**, 1437–1474 (2008).
- [18] Noë, U., Taschler, B., Täger, J., Heutink, P. & Mukherjee, S. Ancestral causal learning in high dimensions with a human genome-wide application. *arXiv preprint arXiv:1905.11506* (2019).
- [19] Alon, U. *An introduction to systems biology: design principles of biological circuits* (CRC press, 2019).
- [20] Hyttinen, A., Eberhardt, F. & Hoyer, P. O. Learning linear cyclic causal models with latent variables. *The Journal of Machine Learning Research* **13**, 3387–3439 (2012).
- [21] Eberhardt, F. & Scheines, R. Interventions and causal inference. *Philosophy of Science* **74**, 981–995 (2007).

- [22] Kocaoglu, M., Shanmugam, K. & Bareinboim, E. Experimental design for learning causal graphs with latent variables. In *Advances in Neural Information Processing Systems 30, NIPS* (2017).
- [23] Replogle, J. M. *et al.* Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq. *Cell* (2022).
- [24] Malinsky, D. & Spirtes, P. Estimating bounds on causal effects in high-dimensional and possibly confounded systems. *International Journal of Approximate Reasoning* **88**, 371–384 (2017).
- [25] Chickering, D. M. Optimal structure identification with greedy search. *The Journal of Machine Learning Research* **3**, 507–554 (2002).
- [26] Shimizu, S., Hoyer, P. O., Hyvärinen, A. & Kerminen, A. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research* (2006).
- [27] Bühlmann, P., Peters, J. & Ernest, J. Cam: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics* (2014).
- [28] Colombo, D., Maathuis, M. H., Kalisch, M. & Richardson, T. S. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics* (2012).
- [29] Ke, N. R. *et al.* Learning neural causal models from unknown interventions. *arXiv preprint arXiv:1910.01075* (2019).