Scalable and Flexible Causal Discovery with an Efficient Test for Adjacency

Alan Nawzad Amin [1]   Andrew Gordon Wilson [1]

Abstract

To make accurate predictions, understand mechanisms, and design interventions in systems of many variables, we wish to learn causal graphs from large scale data. Unfortunately the space of all possible causal graphs is enormous so scalably and accurately searching for the best fit to the data is a challenge. In principle we could substantially decrease the search space, or learn the graph entirely, by testing the conditional independence of variables. However, deciding if two variables are adjacent in a causal graph may require an exponential number of tests. Here we build a scalable and flexible method to evaluate if two variables are adjacent in a causal graph, the Differentiable Adjacency Test (DAT). DAT replaces an exponential number of tests with a provably equivalent relaxed problem. It then solves this problem by training two neural networks. We build a graph learning method based on DAT, DAT-Graph, that can also learn from data with interventions. DAT-Graph can learn graphs of 1000 variables with state of the art accuracy. Using the graph learned by DAT-Graph, we also build models that make much more accurate predictions of the effects of interventions on large scale RNA sequencing data.

1. Introduction

Large scale studies have recently collected hundreds of thousands of measurements of thousands of variables and interventions across genetics, microbiology, and healthcare (Van Hout et al., 2020; Regev et al., 2017; Franzosa et al., 2019; Geiger-Schuller et al., 2023; Replogle et al., 2022; Dixit et al., 2016). An algorithm that leverages this data to learn cause and effect must scale to many measurements and variables, flexibly accommodate complex interactions between variables, and make reliable predictions in realistic and large scale settings.

Modern state-of-the-art algorithms frame learning causal relationships as a model selection problem (Chickering, 2002; van de Geer & Bühlmann, 2013). Each model corresponds to a directed acyclic graph representing which variables cause which others. Complex relationships between variables are then modelled using flexible neural networks (Lachapelle et al., 2019; Zheng et al., 2020). The central practical challenge of this approach is the model search, where one needs to explicitly search through the enormous space of all directed acyclic graphs. Recently, a number of gradient-based search methods have scaled this procedure to data of large complex systems (Zheng et al., 2018; Lachapelle et al., 2019; Zheng et al., 2020; Nazaret et al., 2023). However, these heuristic search procedures can be unstable in practice and can be unreliable even in simple settings (Wei et al., 2020; Nazaret et al., 2023; Deng et al., 2023). The model search also becomes exponentially harder as the number of variables increases.

To make more accurate predictions at scale we can shrink the search space, or avoid searching by learning the graph entirely, by taking the alternative approach of many classical graph learning algorithms. These approaches test the data for conditional independence relationships, and then exclude the corresponding edges from the graph (Spirtes et al., 1993). Indeed recently, by reducing the model search space with some limited testing, a gradient-based model search strategy demonstrated large gains in accuracy on data of large complex systems (Nazaret et al., 2023). And historically, by reducing the search space by testing as much as possible before performing model search, classical non-flexible "hybrid" causal models achieved state-of-the-art accuracy learning from data of simple systems (Tsamardinos et al., 2006; Bühlmann et al., 2014).

Unfortunately, classical testing-based procedures struggle to scalably and reliably learn from data of large complex systems. A particularly informative test, and the central step in virtually all classical testing-based procedures, is evaluating if two variables are immediate causes or effects of each other — that is, if they are adjacent in the causal graph (Spirtes et al., 1993; Tsamardinos et al., 2006). Evaluating this relationship for a pair of variables involves searching for a set of other variables that renders them conditionally independent (Fig. 1(a)). For each pair this search can involve an exponential number of tests. Because of the enormous
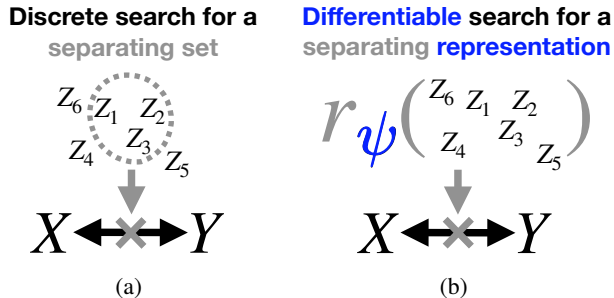
[1]New York University, New York, USA. Correspondence to: Alan Nawzad Amin <alanamin@nyu.edu>.

Figure 1. **Our Differentiable Adjacency Test (DAT) relaxes a discrete search requiring exponentially many tests (a) into a differentiable search to solve an optimization problem (b).**

number of tests, and the particularly large computational cost of flexible conditional independence tests (Zhang et al., 2011; Sen et al., 2017; Berrett et al., 2019; Bellot & van der Schaar, 2019), this procedure does not scale to data of large complex systems. Furthermore, because realistic data can include many spurious borderline conditional independence relations — so-called "violations of faithfulness" — testing for adjacency can be unreliable even in simple settings (Uhler et al., 2012; Andersen, 2013).

Here we develop a testing-based graph learning procedure, DAT-Graph, that can scalably, flexibly, and reliably learn cause and effect from data of large and complex systems. DAT-Graph is centred around a method for evaluating whether two variables are adjacent in the causal graph, DAT. To do this flexibly and scalably, DAT replaces performing an exponential number of tests (Fig. 1(a)) with optimizing a single differentiable objective (Fig. 1(b)). We build DAT carefully so that the testing and optimization problems are provably equivalent. DAT-Graph learns the graph in a way that is reliable even when there are some violations of faithfulness; in particular, since DAT is scalable, DAT-Graph can evaluate the adjacency of every pair of variables with two complementary tests in a way that is too computationally expensive for previous methods. Empirically, we show that DAT-Graph is able to easily scale to learn on real and synthetic data with 10^3 variables and 10^4 observations. We show DAT-Graph learns large sparse causal graphs more accurately than state of the art gradient-based model selection procedures with and without interventions. We also use DAT-Graph to reduce the search space of model selection procedures to build even more accurate hybrid models. We show that these hybrid models accurately predict the effects of interventions on large scale RNA sequencing data.

Our code is available at https://github.com/AlanNawzadAmin/DAT-graph/.

2. Related work

There is an enormous history of methods to learn graphs in general. We give a full review of these methods in Section A.

Efforts to build testing-based graph learning procedures for data of large and complex systems have focused on building more flexible conditional independence tests using kernels (Fukumizu et al., 2007; Zhang et al., 2011; Pogodin et al., 2022), simulation of conditional distributions (Doran et al., 2014; Sen et al., 2017; Berrett et al., 2019; Bellot & van der Schaar, 2019), or a different strategy (Shah & Peters, 2020; Polo et al., 2023; Laumann et al., 2023); or building more scalable tests (Strobl et al., 2019; Runge, 2018) or reducing the number of tests by performing them in a clever order (Margaritis & Thrun, 1999; Mokhtarian et al., 2020). While these methods are accurate at small scale (Pogodin et al., 2022), they unfortunately cannot scale to more than 100 variables. DAT instead replaces an exponential number of tests with an equivalent relaxed problem.

The relaxed problem DAT considers is equivalent to the problem considered in Invariant Risk Minimization (IRM) (Arjovsky et al., 2019) when one of the variables is a discrete environment variable. DAT however can accommodate continuous variables. DAT also diverges strongly from IRM methodologically so that the answer to the relaxed problem is provably identical to the initial problem.

3. Background

We wish to learn causal relationships between N real random variables $X^1, \ldots, X^N$. We model cause and effect relationships; that is, if we call $\mathrm{Pa}(X^n)$ the direct causes of the variable X^n then we assume there is some function h_n such that

$$X^n = h_n\left(\epsilon_n, (X^m)_{m\in\mathrm{Pa}(X^n)}\right) \qquad (1)$$

where $(\epsilon_n)_{n=1}^N$ are iid random variables independent from each other and $X^{1:N}$. We represent causal relations in a graph G with nodes $X^{1:N}$ where there is a directed edge from X^n to X^m if $X^m \in \mathrm{Pa}(X^n)$, that is, each variable X^n is caused by its parents in G, $\mathrm{Pa}_G(X^n)$. We assume that there are no cycles in this graph.

Now we wish to recover the graph G from some observations $X_1^{1:N}, \ldots, X_K^{1:N}$. We assume the observations come iid from some distribution p, and defer observations with intervened variables until Section 6; we call this the "purely observational" setting. First note that in this setting the graph G is only identifiable up to an equivalence class (Pearl, 2010). However, the skeleton of the graph – a graph in which two nodes are connected by an undirected edge if they are adjacent in G – is identifiable (Pearl, 2010), and in many cases we can distinguish cause and effect for almost all adjacent variables in G (Katz et al., 2019). Our goal is to learn a member of the equivalence class of G.

The causal relationships in Eqn 1 necessarily imply observable conditional independence relationships. If the variables $X^{1:N}$ follow Eqn 1 then it must be the case that X^n, when conditioned on its parents $\{X^m\}_{m\in\mathrm{Pa}_G(X^n)}$, is indepen-

dent of all nodes other than its decendents in G, $\text{De}_G(X^n)$:

$$X^n \perp\!\!\!\perp \{X^m\}_{m \notin \text{De}_G(X^n)} \mid \{X^m\}_{m \in \text{Pa}_G(X^n)}. \quad (2)$$

These relationships in turn necessarily imply that any two sets of nodes $A, B \subset X^{1:N}$ that are d-separated in G by a third set of nodes $C \subset X^{1:N}$ are conditionally independent $A \perp\!\!\!\perp B \mid C$ (Geiger et al., 1990) (see Spirtes et al. (1993) for a review of d-separation). In the generic case, these are all of the conditional independence relationships of the nodes $X^{1:N}$, meaning that p is faithful to the graph G (Geiger et al., 1990; Uhler et al., 2012).

Definition 3.1. A distribution p over $X^{1:N}$ is faithful to a graph G if for any three disjoint sets $A, B, C \subset X^{1:N}$, $A \perp\!\!\!\perp B \mid C$ if and only if C d-separates A and B in G.

Our goal is to look for these conditional independence relationships and thereby learn about the topology of the causal graph. Note however that even when p is faithful to a graph G, there are in practice many subsets A, B, C that nearly violate faithfulness – that is, C may not d-separate A and B but A is "almost" independent of B when conditioned on C (Uhler et al., 2012; Andersen, 2013). Therefore, to build a reliable method we would like to be robust to the presence of a few near violations of faithfulness.

A crucial observation is that two non-adjacent nodes in a directed acyclic graph can always be d-separated by some other set of nodes, while adjacent nodes can never be d-separated. Thus if p is faithful to G then two nodes X^n, X^m are adjacent if and only if there is no other set $S \subset \{1, \ldots, N\} \setminus \{n, m\}$ such that $X^n \perp\!\!\!\perp X^m \mid \{X^k\}_{k \in S}$. Thus we can learn the skeleton of G if we had a method to solve what we call the separating set selection problem.

Problem 3.2. (Separating set selection Fig. 1(a)) Given a set of real random variables $X, Y, Z_1, \ldots, Z_M$, is there a subset $S \subset \{1, \ldots, M\}$ such that $X \perp\!\!\!\perp Y | \{Z_m\}_{m \in S}$?

If $\{Z_m\}_{m \in S}$ is any subset such that $X \perp\!\!\!\perp Y | \{Z_m\}_{m \in S}$ then we call it a separating set of X, Y: $\text{SepSet}(X, Y) = \{Z_m\}_{m \in S}$. If p is faithful to G, then if we have the skeleton of G and $\text{SepSet}(X^n, X^m)$ for any pair of nonadjacent nodes X^n, X^m then we can determine the equivalence class of G according to a set of rules (Spirtes et al., 1993). Thus, to learn G from data of large complex systems all we need is a scalable method to solve the separating set selection problem that can flexibly represent complex relationships between $X, Y, Z_1, \ldots, Z_m$.

4. The differentiable adjacency test (DAT)

In this section we build a scalable, flexible, and reliable method to solve the separating set selection problem. In Section 4.1 we first relax the separating set selection problem to a differentiable problem – the separating representation search problem. We prove conditions under which the relaxed problem answers the separating set selection problem. In Section 4.2 we prove that the separating set selection

problem is NP-Hard so we unfortunately cannot guarantee that there are not cases where it is challenging to answer the relaxed problem. Nevertheless, in Section 4.3 we build the Differentiable Adjacency Test (DAT), a practical method to efficiently approximately answer the separating representation search problem using neural networks to represent complex relations between variables. Finally in Section 4.4 we show the DAT solves the separating set problem as accurately as a classical testing approach while being orders of magnitude faster. Throughout we use the term "testing" in the informal sense of a decision rule, without implying validity or coverage.

4.1. Relaxing the discrete search

The computational challenge of the separating set selection problem is the discrete search over all 2^M subsets of $\{1, \ldots, M\}$. We can relax the problem by replacing a search over subsets of $Z_{1:M}$ with a search for a representation $r_{\psi^*}(Z_{1:M})$ where $\{r_\psi\}_{\psi \in \Psi}$ is a differentiably parameterized family of functions that have domain $\mathbb{R}^M$.

Problem 4.1. (Separating representation search Fig. 1(b)) Given a set of real random variables $X, Y, Z_1, \ldots, Z_M$ and a class of possibly random functions $\{r_\psi\}_{\psi \in \Psi}$, is there a $\psi^* \in \Psi$ such that $X \perp\!\!\!\perp Y | r_{\psi^*}(Z_{1:M})$?

Indeed, the separating representation search problem is a natural relaxation of the separating set selection problem that has come up in other causal inference settings in the case that Y is a discrete "environment" variable (Arjovsky et al., 2019; Shi et al., 2020).

In our setting we need to pick our representations $\{r_\psi\}_{\psi \in \Psi}$ so that 1) there is a separating representation if and only if there is a separating set and 2) we can get a separating set $\text{SepSet}(X, Y)$ from the separating representation parameter ψ^*. Unfortunately, there are seemingly reasonable choices in simple situations in which these desiderata are not fulfilled.

Example 4.2. (Existence of a separating representation but no separating set) There are jointly Gaussian variables X, Y, Z_1, Z_2 that are faithful to some graph such that $X \not\perp\!\!\!\perp Y | \{Z_m\}_{m \in S}$ for any $S \subset \{1, 2\}$ but if $\{r_\psi\}_{\psi \in \Psi}$ is the space of linear functions, there is a ψ^* such that $X \perp\!\!\!\perp Y | r_{\psi^*}(Z_{1:2})$.

Proof. In Appendix E.3. □

Unfortunately, by relaxing the problem, we have in effect increased the number of opportunities for there to be a violation of faithfulness from an exponential number – checking $X \perp\!\!\!\perp Y | \{Z_m\}_{m \in S}$ for all subsets S – to an infinite number – checking $X \perp\!\!\!\perp Y | r_\psi(Z_m)_{m=1}^M$ for all ψ.

To build a method with our desiderata, our strategy will be to restrict what $\{r_\psi\}_\psi$ can represent. We choose r_ψ to only represent "soft" subsets of $Z_{1:M}$ where each variable Z_m is softly included in the separating set by mixing it

with independent noise. Let $N_1, \ldots, N_M$ be drawn independently from distributions with densities $f_1, \ldots, f_M$. For $\psi = (\psi_1, \ldots, \psi_M)$ with $\psi_m \in [0, 1]$, we define the noised variable

$$\tilde{Z}_{\psi,m} = \psi_m Z_m + (1 - \psi_m) N_m$$

and the representation $r_\psi(Z_{1:M}) = \tilde{Z}_{\psi,1:M}$. By mixing Z_m with an independent random variable N_m we lose information about Z_m. ψ_m controls how much information we observe about Z_m; when $\psi_m = 0$ we do not observe Z_m and when $\psi_m = 1$ we observe Z_m fully.

With this choice, one can still design $f_1, \ldots, f_M$ to get disagreeing answers between the separating set selection and separating representation search problems (see Example E.4). However we prove that if we pick $f_1, \ldots, f_M$ to have thick tails then the answer to the two problems will be identical and we can recover a separating set from a separating representation as desired.

Theorem 4.3. *(Proof in Appendix E.4) Assume Assumption E.6 (pick f_m to have thicker tails than p). The separating set selection problem and the separating representation search problem have the same answer. If $\tilde{Z}_{\psi^*,1:M}$ is a separating representation then $\{Z_m\}_{\psi_m^* = 1}$ is a separating set.*

4.2. Hardness of adjacency testing

In Thm. 4.3 we showed that if we find a separating representation with parameter ψ^* then we obtain a separating set. Can we build an efficient method in practice that is guaranteed to find ψ^*? We answer this question negatively.

Proposition 4.4. *(Proof in Appendix E.1) Even when restricted to the case where $X, Y, Z_1, \ldots, Z_M$ are jointly Gaussian with known non-singular covariance matrix, the separating set selection problem is NP-Hard.*

There may therefore be cases where, by failing to find ψ^*, we may incorrectly determine that two variables are adjacent in G. Nevertheless, we aim to provide a method that gives accurate approximate solutions with reasonable compute.

4.3. Differentiable Adjacency Test

We now build a method to search for a separating representation in practice. Our method, DAT, will perform this search by minimizing a differentiable objective.

To replace the separating representation search problem with a differentiable optimization problem, we need to replace $X \perp\!\!\!\perp Y | \tilde{Z}_\psi$, where we write $\tilde{Z}_\psi = \tilde{Z}_{\psi,1:M}$, with a differentiable objective $L(\psi)$ that reaches its minimum if and only if $X \perp\!\!\!\perp Y | \tilde{Z}_\psi$. While a number of flexible measures of conditional independence exist (Zhang et al., 2011; Pogodin et al., 2022; Bellot & van der Schaar, 2019), we pick $L(\psi)$ to be the "variance of X explained by Y when conditioned on $\tilde{Z}_\psi$" as it is a well studied measure of conditional independence (Zhang & Janson, 2020; Polo et al., 2023) that is easy to optimize in practice and allows us to model the

relationships between variables with scalable and flexible neural networks:

$$E\mathbb{V}(X; Y | \tilde{Z}_\psi) =$$
$$E[X - E[X | \tilde{Z}_\psi]]^2 - E[X - E[X | Y, \tilde{Z}_\psi]]^2.$$

This quantity is always non-negative and if $X \perp\!\!\!\perp Y | Z$ then it is 0. On the other hand, if $E\mathbb{V}(T(X); Y | \tilde{Z}_\psi) = 0$ for all bounded functions T, then $X \perp\!\!\!\perp Y | Z$. We increase the ability of this metric to detect that $X \not\perp\!\!\!\perp Y | \tilde{Z}_\psi$ by evaluating the variance explained of more than one statistic of X, $T_1(X), \ldots, T_C(X)$,

$$\sum_{c=1}^{C} E\mathbb{V}(T_c(X); Y | \tilde{Z}_\psi). \tag{3}$$

In experiments we choose $C = 2$ and T_1, T_2 as the first and second moments of X. For clarity we write the rest of the section as if $C = 1$ and $T_1(X) = X$.

The variance explained cannot be exactly calculated, so we must approximate it. We first approximate $E[X | \tilde{Z}_\psi]$ with a neural network $g_{\theta_1} : \mathbb{R}^M \to \mathbb{R}$ by training it to minimize the objective

$$L_1(\theta_1) = E\left[X - g_{\theta_1}(\tilde{Z}_\psi)\right]^2.$$

Then, calling the residue $R = X - E[X | \tilde{Z}_\psi] \approx X - g_{\theta_1}(\tilde{Z}_\psi)$, we approximate $E[R | Y, Z]$ with a neural network $g_{\theta_2} : \mathbb{R}^{M+1} \to \mathbb{R}$ by training it to minimize the objective

$$L_2(\theta_2) = E\left[X - g_{\theta_1}(\tilde{Z}_\psi) - g_{\theta_2}(Y, \tilde{Z}_\psi)\right]^2.$$

Finally we optimize ψ to minimize the approximate variance explained,

$$E\mathbb{V}(X; Y | \tilde{Z}_\psi) =$$
$$= E[X - E[X | \tilde{Z}_\psi]]^2 - E[R - E[R | Y, \tilde{Z}_\psi]]^2$$
$$\approx E[X - g_{\theta_1}(\tilde{Z}_\psi)]^2 - E[X - g_{\theta_1}(\tilde{Z}_\psi) - g_{\theta_2}(Y, \tilde{Z}_\psi)]^2$$
$$= L_{\text{DAT}}(\psi).$$

We call this optimization problem the Differentiable Adjacency Test (DAT). We optimize by gradient descent, alternately updating θ_1, θ_2, ψ by taking gradient steps $\nabla_{\theta_1} L_1, \nabla_{\theta_2} L_2, \nabla_\psi L_{\text{DAT}}$ and approximating the expectations with mini-batches of the data. For every data point in a mini-batch we draw independent noise $N_1, \ldots, N_M$ to calculate $\tilde{Z}_\psi$. Training can be framed as a two player game between $\{\theta_1, \theta_2\}$ and ψ, which is not guaranteed to converge to an optimum. In practice however, this is not a challenging optimization problem as the number of variables of one of the players $\psi_{1:M}$ is small – we do not observe the challenges with optimizing multiplayer games such as cycles or instability.

Once we have finished training $\theta_1^*, \theta_2^*, \psi^*$, in theory we need to evaluate if $L_{\text{DAT}}(\psi^*) = 0$ and choose $\text{SepSet}(X, Y) = \{Z_m\}_{\psi_m = 1}$ if so. In practice, we first choose two thresholds $\eta_1, \eta_2 > 0$, and we use a mini-batch of data to get an approximation of the variance explained $\hat{L}_{\text{DAT}}(\psi^*)$. Then

we decide $X \perp\!\!\!\perp Y | \tilde{Z}_{\psi^*}$ if and only if $\hat{L}_{\text{DAT}}(\psi^*) < \eta_1$ and we set $\text{SepSet}(X, Y) = \{Z_m\}_{\psi_m > \eta_2}$

## 4.4. DAT is accurate and efficient in practice

Before using the DAT to learn an entire graph, we evaluate its ability to solve the separating set selection problem. We compare the DAT with the classical method to solve the separating set selection problem: test whether X is independent of Y given S for each subset S of $Z_{1:M}$ and conclude that there is a separating set if the maximum of a test statistic across all tests is above a threshold; then infer that the S for which the test statistic is maximized is a separating set.

We generated small Erdos-Renyi random graphs such that each node had an average of one parent and selected X and Y to be two random nodes and $Z_{1:M}$ to be all other nodes. We then generated 10000 data points from the graph with two layer neural networks as functional relationships as described in Section 7. We consider five conditional independence tests: **RCoT** (Strobl et al., 2019): a scalable and flexible kernel test for conditional independence. **GCIT** (Bellot & van der Schaar, 2019): A flexible simulation-based conditional independence testing method; it learns a conditional distribution by training a GAN. **CCIT** (Sen et al., 2017): A scalable and flexible simulation based independence testing method; it simulates a conditional distribution by a nearest-neighbors search. **CMIT** (Runge, 2018): A flexible method based on estimating conditional mutual information by looking for nearest neighbors. **AT_discrete**: A method to test the conditional independence of two variables by estimating the conditional variance explained; it is exactly DAT without a differentiable search. Due to compute limitations, we were only able to run GCIT, CMIT, and AT_discrete to $M = 3$, while CCIT could scale to $M = 7$ and RCoT to $M = 11$.

We benchmark the accuracy of each method in classifying adjacent variables by the Area Under the receiver operator Curve (AUC) of the test statistic. In Fig. 2(a) (left), we show that DAT is nearly as accurate as state of the art conditional independence tests in determining if two variables are adjacent. In Fig. 2(a) (right) we see that DAT also identifies a separating set nearly as accurately as classical methods.

In Fig. 2(b) (left) we show that all testing methods scale exponentially in compute with M while DAT does not[1]. Finally we show that the scalability of DAT enables us to accurately learn large graphs. We extrapolate the exponential scaling of Fig. 2(b) (right) of the most scalable test, RCoT, and project how much time it would take to learn large graphs swapping DAT with RCoT in our experiments

---

[1] In theory DAT scales linearly with M with a large constant overhead for data transfers to the GPU; for small M however, DAT can parallelized testing many edges at once on a GPU and reduce the computation per edge.

(a) DAT accuracy

(b) DAT efficiency
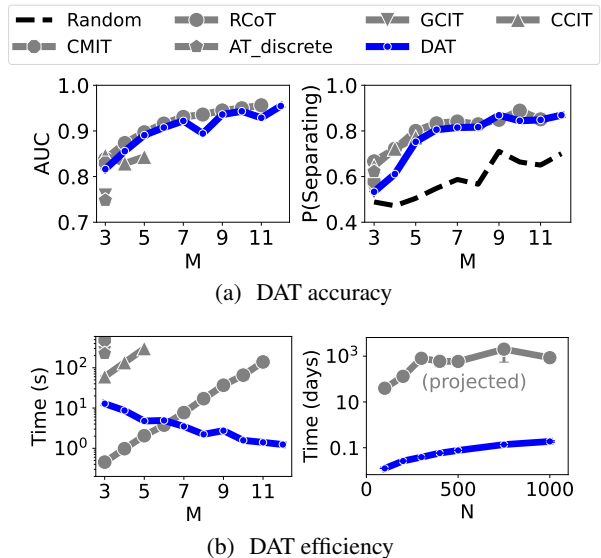
Figure 2. **DAT enables learning large graphs by solving the separating set selection problem accurately and efficiently.** (a) We plot how accurately each method determines if two variables are adjacent (AUC) and how often it correctly identifies a separating set for two non-adjacent variables (P(Separating)) against number of variables (M). (b) We plot the time of running each method against number of variables (M) and the time it would take to use each method to learn a large graph against the size of a graph (N). We plot the mean and standard error across 3 replicates.

in Section 7 (here M could reach 30). We see in Fig. 2(b) (right) that it would take weeks or years to learn graphs with many variables N with classical tests while DAT took minutes to hours in our experiments.

5. Learning a graph with DAT (DAT-Graph)

In this section we use DAT to build a scalable, flexible, and reliable method to learn a causal graph G from data of large, complex systems. We call this method DAT-Graph.

In principle we could learn the graph by testing the adjacency of every pair of nodes using DAT. This would involve N^2 tests, each of which involves a search over $N - 2$ variables. This strategy is clearly not scalable, and is also unreliable under possible near violations of faithfulness. Instead we use a strategy employed by a large number of hybrid and testing-based graph-learning methods to first efficiently and reliably exclude a large number of edges from G (Margaritis & Thrun, 1999; Mokhtarian et al., 2020; Bühlmann et al., 2014; Nazaret et al., 2023). The idea is to try and predict the distribution of each variable X^n using all other variables – if we then determine that a variable X^m is not useful in predicting X^n we can conclude that it cannot be connected to X^n in G. Then we only need to solve the separating set selection problem to test and orient the substantially reduced number of remaining edges. This first step also reduces the search space of each separating set selection problem (Margaritis & Thrun, 1999; Mokhtarian et al., 2020).

However following previous methods by performing just these tests still leaves DAT-Graph unreliable when there are near violations of faithfulness. To greatly increase its reliability in theory and in practice, unlike previous methods, DAT-Graph evaluates the adjacency of each pair of nodes by solving two separating set selection problems with different search spaces; it is able to perform both tests efficiently due to the scalability of DAT.

In Section 5.1 we describe how we learn the moral graph – the graph that represents which variables are useful for predicting which others. In Section 5.2 we describe how learning the moral graph reduces the search space for each adjacency test; we also describe how we perform DAT twice to test the adjacency of each pair of variables. In Section 5.3 we discuss the computational complexity of DAT-Graph.

Once we have learned the skeleton of G, we need to orient its edges. We review how to do so using the separating sets we have calculated from DAT and standard rules from classical testing methods (Spirtes et al., 1993) in Appendix B.1.

5.1. Learning the moral graph

The first step in DAT-Graph is to exclude edges between variables that are not useful in predicting each other. To do so, we must identify, for each variable X^n, the sparsest set in $\{X^m\}_{m \neq n}$ that can predict the distribution of X^n. This sparsest set is known as the Markov blanket of X^n.

Definition 5.1. A Markov blanket of a variable X^n is a smallest subset $\mathrm{MB}(X^n) \subset \{X^m\}_{m \neq n}$ that makes X^n conditionally independent of all other variables

$$X \perp\!\!\!\perp \{X^m\}_{m \neq n} \setminus \mathrm{MB}(X^n) \mid \mathrm{MB}(X^n).$$

The moral graph is the graph with an undirected edge between variables X^n and X^m if $X^n \in \mathrm{MB}(X^m)$. When p is faithful, the Markov blanket of a variable X^n is a unique set consisting of all the variables that are adjacent to X^n in G as well as "spouses" of X^n – variables that are not adjacent to X^n in G but share a direct child with X^n (Spirtes et al., 1993). Thus edges that are not in the moral graph also are also not in the skeleton of G. If we can learn the moral graph we can therefore exclude many edges from G.

Schmidt et al. (2007), Bühlmann et al. (2014), and Nazaret et al. (2023) have shown that the Markov blanket of X^n is the solution to any sparse variable selection procedure. To solve sparse variable selection there are an enormous number of scalable, flexible, and reliable algorithms. We adapt a variable selection method from Nazaret et al. (2023) that allows us to flexibly model causal relationships using neural networks. For each n we predict X^n using all other variables $\{X^m\}_{m \neq n}$ using a neural network $g_{\theta_n} : \mathbb{R}^{N-1} \to \mathbb{R}$. We encourage g_{θ_n} to predict the expectation of X^n using a sparse set of variables in $\{X^m\}_{m \neq n}$ by L1-regularizing the weights of its first layer. In particular, calling W^n the

first layer weights of the network, g_{θ_n} is trained to minimize the objective

$$L_n(\theta_n) = E\left[X^n - g_{\theta_n}(X^m)_{m \neq n}\right]^2 + \lambda \sum_{i,j} |W_{i,j}^n|.$$

After training g_{θ_n} we can get a measure of the importance of X^m in predicting X^n as $\alpha_{X^n, X^m} = \sum_i (W_{i,m}^n)^2$. To get a moral graph in practice, we pick a threshold η_3 and connect X^n and X^m if $\alpha_{X^n, X^m} + \alpha_{X^m, X^n} > \eta_3$, that is, if X^n is predicted to be in $\mathrm{MB}(X^m)$ or X^m is predicted to be in $\mathrm{MB}(X^n)$.

It is possible that g_{θ_n} may erroneously ignore a variable X^m which affects X^n without changing its expectation. To avoid this, as in Section 4.3, we can train g_{θ_n} to predict the expectation of multiple statistics $T_1(X^n), \ldots, T_C(X^n)$. Again, in experiments we choose $C = 2$ and T_1, T_2 as the first and second moments of X.

5.2. Testing adjacency with two DATs

With knowledge of the moral graph, we can reduce the number of adjacency tests we must perform to learn the skeleton of G. We can also reduce the search space for each test we perform: the following proposition adapted from from Margaritis & Thrun (1999) states that to test if two variables are adjacent in G, instead of searching for a separating set in the set of all other variables, we can restrict our search to the Markov blanket of one of the variables.

Proposition 5.2. (*Margaritis & Thrun, 1999*) (Proof in Appendix E.2) Assume p is faithful. X^n and X^m are adjacent in G if and only if $X^n \not\perp\!\!\!\perp X^m \mid U$ for any $U \subset \mathrm{MB}(X^n)$.

Now, to test if X^n is adjacent to X^m we can choose to search for a separating set $U \subset \mathrm{MB}(X^n) \setminus \{X^m\}$ or for a separating set $U \subset \mathrm{MB}(X^m) \setminus \{X^n\}$. Classical testing methods solve the separating set selection problem by testing every subset of the search space; thus these methods saved a large amount of compute by learning the moral graph and then only solving the separating set selection problem corresponding to the smaller Markov boundary (Margaritis & Thrun, 1999; Mokhtarian et al., 2020).

DAT does not scale poorly with the size of the search space, so DAT-Graph instead performs both tests and concludes that X^n adjacent to X^m if either of the tests states they are adjacent. Performing two tests increases the reliability of the adjacency test: there may be some subset $U \subset \mathrm{MB}(X^n) \setminus \{X^m\}$ that violates faithfulness with $X^n \not\perp\!\!\!\perp X^m | U$ but such that U is not a subset of $\mathrm{MB}(X^m) \setminus \{X^n\}$.

Example 5.3. (Performing two tests increases reliability) (Proof in Appendix E.3) There are four jointly Gaussian random variables that are not faithful such that DAT-Graph recovers the correct graph.

Performing two tests also follows the idea of other methods that perform multiple tests that are redundant in the faithful

case to be reliable when there are violations of faithfulness (Spirtes & Zhang, 2014; Marx et al., 2021).

In practice, we take the statistics of the two tests from Section 4.3, $\hat{L}_{\text{DAT}}^n(\psi^*)$ and $\hat{L}_{\text{DAT}}^m(\psi^*)$, and decide that X^n is adjacent to X^m if either test statistic is large: $\hat{L}_{\text{DAT}}^n(\psi^*)^2 + \hat{L}_{\text{DAT}}^m(\psi^*)^2 > \eta_1$.

5.3. Computational cost of DAT-Graph

DAT-Graph has two steps. In its first step it learns the moral graph. This in principle scales quadratically with N, but in practice can scale to $N = 10^4$ in hours (Nazaret et al., 2023). In the second step, DAT-Graph performs two adjacency tests for every edge in the moral graph. If s is the average number of parents in the graph G then the Markov blanket of a variable X^n often has approximately $O(s^2)$ edges. Thus DAT-Graph needs to perform $O(Ns^2)$ tests which each involve a search over $O(s^2)$ variables. Thus in principle, this step of DAT-Graph scales linearly with N and is much faster on sparser graphs.

6. Learning from data with interventions

In the purely observational setting, the graph G can only be determined up to an equivalence class – for some variables, we cannot distinguish which is the cause and which is the effect. To learn cause and effect for these variables, we can collect data in an experiment where we intervene on a variable (Hauser & Bühlmann, 2012). For example, we may knock down a gene in a cell. We can then distinguish between cause and effect as intervening on a cause should affect the distribution of an effect, but not vise-versa.

In this section we extend DAT-Graph to learn from observational as well as intervention data of large and complex systems. To do so, we model an intervention on a variable X^n as a change in the dependence on its parents h_n as defined in Eqn. 1. We can represent this by adding an extra argument to h_n

$$X^n = h_n\left(I, \epsilon_n, (X^m)_{m \in \text{Pa}(X^n)}\right), \qquad (4)$$

where I is a binary variable representing the presence of an intervention on X^n. In intervention data we may have a number of intervention targets $X^{m_1}, \ldots, X^{m_K}$ with intervention indicators $I^1, \ldots, I^K$. We assume we know which variables are intervened upon in each experiment, so $I^1, \ldots, I^K$ are observed.

Just as Eqn. 1 produces a causal graph G over the variables $X^1, \ldots, X^N$, Eqn. 4 produces a causal graph over the extended set of variables $X^1, \ldots, X^N, I^1, \ldots, I^K$ that has G as an induced sub-graph. We can therefore learn from intervention data by applying our method to learn the graph over the extended set of variables. This approach is known as joint causal inference (Mooij et al., 2020).

Including the interventions in inference can help in orienting edges of the graph (Mooij et al., 2020). We however note that if the targets of the intervention are known, then intervention data can also help learn the skeleton, reduce the number of tests we must perform, and can reduce the search space of our tests. In Section B.2 we describe our method to learn from intervention data with known targets and show it help learn graphs more accurately in theory and in practice.

7. Experiments

Here we demonstrate that DAT-Graph can accurately and scalably learn graphs from data of large complex systems, with or without interventions, and on real and synthetic data. We show that DAT-Graph performs particularly well on sparser graphs. We also show that we can also combine the strengths of DAT-Graph and gradient-based model search methods in a hybrid method.

We measure the accuracy of inferred graphs with the Structural Hamming Distance (SHD) between the inferred skeleton and the true skeleton, and the SHD of the inferred directed graph and the true directed graph. The SHD of the skeletons is the number of incorrectly inferred edges. The SHD of the directed graphs is the SHD of the skeletons plus the number of incorrectly directed edges.

To implement DAT, we need to pick the distribution f_m of the noise variables $(N_m)_m$ from Section 4.1. To sample N_m we first take $\tilde{N}_M \sim$ Laplace; if $|\tilde{N}_M| \leq 1$ then we set $N_m = \frac{1}{2}\tilde{N}_m$, otherwise, we scale $\tilde{N}_m$ to get thicker tails: $N_m = \frac{1}{2}\text{sgn}(\tilde{N}_m)|\tilde{N}_m|^{1.1}$. In Appendix E.4.1 we show that this choice satisfies the assumptions of Thm. 4.3. We perform all experiments on a single CPU and a single RTX 8000 GPU. Other details of DAT are described in Appendix B. Experimental details are described in Appendix C.

7.1. Learning from observational data

Setup First we demonstrate that DAT-Graph can accurately learn a causal graph from data of large complex systems in the purely observational setting. To compare methods as fairly as possible, we simulate data of large complex systems as reported in the state of the art gradient-based model search method, SDCD, from Nazaret et al. (2023). To do so, we generate an Erdős-Renyi random directed graph G over N variables with s average parents. We simulate complex relations between variables by using randomly initialized two layer neural networks with additive Gaussian noise as the causal relations between variables in Eqn. 1. We then generate 10000 datapoints from this model.

Nazaret et al. (2023) performed a thorough investigation of the scalability of different graph learning algorithms. They showed that, on this data, existing algorithms – other than their algorithm, SDCD – do not scale to more than 100 variables or only achieve trivial accuracy. We therefore use
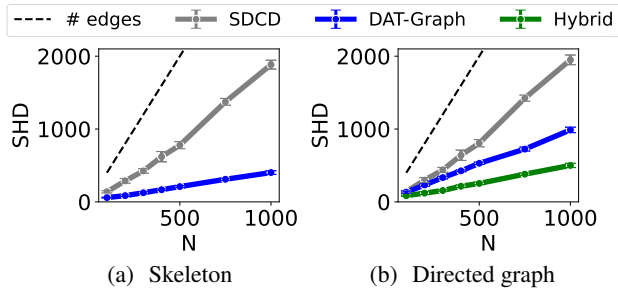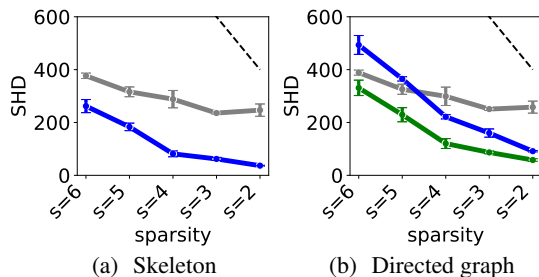
Figure 3. **DAT-Graph learns large graphs accurately.** We plot the mean error (SHD) and standard error against the size of the graph (N) across 3 replicates.

Figure 4. **DAT-Graph learns sparser graphs more accurately.** We plot the mean and standard error against the sparsity of the graph across 3 replicates. The legend is the same as that of Fig. 3.

SDCD as our baseline. SDCD is a gradient-based model search method with a similar first step to DAT-Graph – they both first learn the moral graph using a variable selection procedure. Then, when SDCD is learning the causal graph, it masks edges that are not in the learned moral graph.

Large graphs   In Fig. 3 we plot the SHD in the inferred skeleton and inferred directed graph on data with $s = 4$ and various values of N as in Nazaret et al. (2023). We note DAT-Graph infers the graph just as accurately as the state of the art method SDCD at all N. We also note that DAT-Graph becomes relatively more accurate as N increases, in this case beating the state of the art model SDCD by a substantial margin for large N. In Fig. 7 and 8 in the Appendix we also show a similar result when the graph is generated from a scale-free distribution or with linear relations. In App.D.2 we show that DAT-Graph achieves state of the art accuracy among small-scale methods as well.

Hybrid model   Next we demonstrate that DAT-Graph and model search methods have complimentary strengths that can be combined in a hybrid model. The SHD of SDCD's inferred skeleton is similar to the SHD of its directed graph, meaning it often picks the correct direction for arrows in the graph. This is not the case for DAT-Graph, which infers a very accurate skeleton but makes more errors when orienting edges. To combine the advantages of these methods, we create a hybrid method by first learning a skeleton using DAT-Graph and then learning the directions of the edges using SDCD. Fig. 3(b) shows that this hybrid method performs substantially better than both methods at scale.

Scaling   Our results also demonstrate that DAT-Graph can theoretically scale to learn from very large datasets. SDCD is incredibly scalable – it infers a graph of 1000 nodes in roughly 25 minutes. DAT-Graph is not as scalable as SDCD however it scales to large systems in reasonable time – it infers a graph of 1000 nodes in roughly 4 and a half hours. As well, in Fig. 9(a) we show that compute time for DAT-Graph scales roughly linearly with N; this is expected as, keeping s fixed, the number of adjacency tests in DAT-Graph scales linearly with N as discussed in Section 5.2. If this linear trend continuous, DAT-Graph could learn from

the largest transcriptomics datasets, which could include a variable for all 20000 genes in a human, in a few days.

Sparsity   Graphs of real data are likely to be sparse – each variable is caused by few others – and we would like to take advantage of this sparsity to learn a more accurate graph. In principle, both SDCD and DAT-Graph take advantage of sparsity to exclude more edges when learning the moral graph. In Fig. 4 we test how well each method takes advantage of sparsity in practice by plotting the error in the inferred graphs for datasets with $N = 200$ and various average numbers of parents s. All methods perform more accurate inference on sparser graphs, but DAT-Graph benefits from sparsity much more than SDCD – when $s = 2$ the mean SHD of the graph inferred by SDCD is 112, while that of DAT-Graph and the hybrid method are 45 and 29 respectively. In Fig. 9(b) in the Appendix we also show that DAT-Graph also requires less compute to learn sparser graphs. On the other hand, SDCD makes more accurate predictions on denser graphs, a possible advantage gradient-based learning methods. In Fig. 10 in the Appendix we confirm that the conclusion of Fig. 3(b) do not change when the graph is dense – DAT-Graph and the hybrid method make more accurate predictions as the graph gets larger.

7.2. Learning from intervention data

Given more intervention data, we expect to be able to learn a more accurate graph. To see if DAT-Graph efficiently uses intervention data to learn more accurate graphs, we simulate data similar to the setup above but intervene on certain variables. Intervened variables are drawn from a Gaussian distribution with standard deviation 0.1. We vary the fraction of variables that are intervened upon. We simulate 10000 datapoints from the observational distribution and for every intervened variable we sample another 500 datapoints where that variable is intervened upon. Nazaret et al. (2023) demonstrated that SDCD learns from intervention data substantially better than other methods. Thus we use SDCD as our baseline.

In Fig. 5 we plot the SHD of the skeleton and directed graph when $N = 100$, $s = 4$ for datasets with various fractions of variables intervened. We see that all methods efficiently
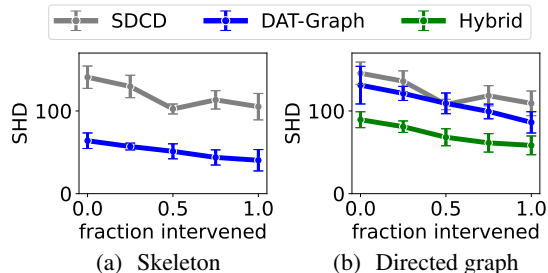
Figure 5. **DAT-Graph learns more accurate graphs when given intervention data.** We plot the mean and standard error against the number of variables with interventions across 5 replicates.

use intervention data – an increasing amount of intervention data makes predictions more accurate.

7.3. Sensitivity to model choices

In Section 4.1 and 5.2 we justified our choices of representation r_ψ and testing each edge twice in theory. In Section C.2 we perform ablations that show that these choices also substantially increase the accuracy of DAT-Graph in practice.

In Fig. 11 and 12 in the appendix we show that DAT-Graph is robust to neural network and threshold hyperparameter choices. In Fig. 13 we investigate the effect of adding more statistics to our estimate of variance explained in Section 4.3; we see including the first two moments does better than including only the first moment or the first three, likely because two moments best balances the ability for us to detect dependencies with the variance of the estimator.

7.4. Predicting interventions on RNA sequencing data

Learning which variables are causes of which others in principle allows us to better predict the effects of interventions. In this section we investigate whether the graph learned by DAT-Graph can be useful for predicting interventions on large complex systems. Here we learn from a single-cell RNA sequencing experiment of cancer that is resistant to immunotherapy (Frangieh et al., 2021) to predict the effects of gene knockdowns. Good prediction can tell us about the mechanisms of resistance and suggest targets for treatment.

Each variable X^n is the normalized transcript count of gene n and each data point is the transcript counts for every gene in a cell. Interventions are CRISPR gene knockdowns; there can be multiple interventions per cell. We preprocessed this data as in Lopez et al. (2022). We first split the data into the three cell populations studied in Frangieh et al. (2021) — control, co-culture, and IFN-γ-treated cells. We then filtered to predict on the $N = 1000$ most variable genes. We split each dataset into a training set and a test set containing interventions that are not in the training set.

We infer a graph from each training set using DAT-Graph and evaluate whether restricting the graph search of SDCD with these graphs can improve prediction. In Fig. 6 we show
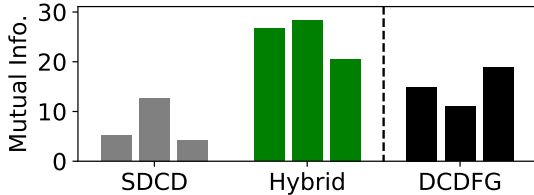

Figure 6. **DAT-Graph helps predict the effects of unseen interventions on RNA sequencing data.** We plot the learned mutual information — the difference between the mean log likelihood of a model and the mean log likelihood of a trivial model with an empty graph on the test set — of models for the datasets "control", "IFN-γ", and "co-culture". We compare two graph search algorithms — SDCD and DCDFG — to our method which is a hybrid of DAT-Graph and SDCD.

that SDCD's prediction improves substantially when graph search is restricted to edges learned in the skeleton of DAT-Graph. In Appendix D.3 we show that this improvement is not an artefact of training SDCD. The hybrid model also outperforms another gradient-based model search method, DCDFG, which was built to learn on large-scale RNA sequencing data (Lopez et al., 2022).

8. Conclusion

We have developed DAT-Graph to scalably and reliably learn cause and effect from data of large and complex systems.

There are many exciting directions for future work. While DAT-Graph accurately learns sparse graphs, it can be less accurate than model selection based methods when learning dense graphs; future work could build hybrid models that have the strengths of both approaches. DAT-Graph also follows a trend of modern multi-step graph learning methods that come with more hyperparameters (Nazaret et al., 2023; Lopez et al., 2022); future work could reduce the number of hyperparameters by combining steps in these methods, for example by reusing neural networks between steps.

In this work we have made the assumption that all variables are observed and there are no cycles in G. However, there are large and complex systems where these assumptions are not to likely to hold (Sethuraman et al., 2023; Lorch et al., 2023). Methods have been developed to test for confounding and cycles by solving the separating set selection problem, but only at small scale (Spirtes, 2001; Richardson, 1996). Future work could scale these methods using DAT.

In this work we have considered learning the entire graph G. However, often one is only interested in learning cause and effect for a particular node. Unfortunately model search methods must learn the whole graph. Future work may apply DAT to only learn the skeleton around this node. In addition to saving compute, learning the skeleton without necessarily orienting edges is a procedure that is robust to confounding or cycles (M. Mooij & Claassen, 2020).

Impact Statement

DAT-Graph allows accurate inference of cause and effect in systems of many variables that interact in complex ways. Such systems appear in genetics, microbiology, and health. Understanding cause and effects in these settings can help us understand the mechanisms of disease, and help us build better interventions and treatments. On the other hand, causal conclusions about genetics and phenotype can be used to justify harmful policies.

References

Andersen, H. When to expect violations of causal faithfulness and why it matters. *Philos. Sci.*, 80(5):672–683, December 2013.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. July 2019.

Bello, K., Aragam, B., and Ravikumar, P. DAGMA: Learning DAGs via m-matrices and a log-determinant acyclicity characterization. *Adv. Neural Inf. Process. Syst.*, 2022.

Bellot, A. and van der Schaar, M. Conditional independence testing using generative adversarial networks. *Adv. Neural Inf. Process. Syst.*, 2019.

Berrett, T. B., Wang, Y., Barber, R. F., and Samworth, R. J. The conditional permutation test for independence while controlling for confounders. *J. R. Stat. Soc. Series B Stat. Methodol.*, 82(1):175–197, October 2019.

Brouillard, P., Lachapelle, S., Lacoste, A., Lacoste-Julien, S., and Drouin, A. Differentiable causal discovery from interventional data. *Adv. Neural Inf. Process. Syst.*, July 2020.

Bühlmann, P., Peters, J., and Ernest, J. CAM: Causal additive models, high-dimensional order search and penalized regression. *aos*, 42(6):2526–2556, December 2014.

Chickering, D. M. Optimal structure identification with greedy search. https://www.jmlr.org/papers/volume3/chickering02b/chickering02b.pdf, 2002. Accessed: 2023-11-25.

Deng, C., Bello, K., Aragam, B., and Ravikumar, P. Optimizing NOTEARS objectives via topological swaps. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *ICML'23*, pp. 7563–7595. JMLR.org, July 2023.

Dixit, A., Parnas, O., Li, B., Chen, J., Fulco, C. P., Jerby-Arnon, L., Marjanovic, N. D., Dionne, D., Burks, T., Raychowdhury, R., Adamson, B., Norman, T. M., Lander, E. S., Weissman, J. S., Friedman, N., and Regev,

A. Perturb-Seq: Dissecting molecular circuits with scalable Single-Cell RNA profiling of pooled genetic screens. *Cell*, 167(7):1853–1866.e17, December 2016.

Doran, G., Muandet, K., Zhang, K., and Schölkopf, B. A permutation-based kernel conditional independence test. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, pp. 132–141, Arlington, Virginia, USA, July 2014. AUAI Press.

Frangieh, C. J., Melms, J. C., Thakore, P. I., Geiger-Schuller, K. R., Ho, P., Luoma, A. M., Cleary, B., Jerby-Arnon, L., Malu, S., Cuoco, M. S., Zhao, M., Ager, C. R., Rogava, M., Hovey, L., Rotem, A., Bernatchez, C., Wucherpfennig, K. W., Johnson, B. E., Rozenblatt-Rosen, O., Schadendorf, D., Regev, A., and Izar, B. Multimodal pooled Perturb-CITE-seq screens in patient models define mechanisms of cancer immune evasion. *Nat. Genet.*, 53 (3):332–341, March 2021.

Franzosa, E. A., Sirota-Madi, A., Avila-Pacheco, J., Fornelos, N., Haiser, H. J., Reinker, S., Vatanen, T., Hall, A. B., Mallick, H., McIver, L. J., Sauk, J. S., Wilson, R. G., Stevens, B. W., Scott, J. M., Pierce, K., Deik, A. A., Bullock, K., Imhann, F., Porter, J. A., Zhernakova, A., Fu, J., Weersma, R. K., Wijmenga, C., Clish, C. B., Vlamakis, H., Huttenhower, C., and Xavier, R. J. Gut microbiome structure and metabolic activity in inflammatory bowel disease. *Nat Microbiol*, 4(2):293–305, February 2019.

Fukumizu, K., Gretton, A., Sun, X., and Schölkopf, B. Kernel measures of conditional dependence. In Platt, J., Koller, D., Singer, Y., and Roweis, S. (eds.), *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

Gao, M., Ding, Y., and Aragam, B. A polynomial-time algorithm for learning nonparametric causal graphs. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, number Article 973 in NIPS'20, pp. 11599–11611, Red Hook, NY, USA, December 2020. Curran Associates Inc.

Geiger, D., Verma, T., and Pearl, J. Identifying independence in bayesian networks. *Networks*, 20(5):507–534, August 1990.

Geiger-Schuller, K., Eraslan, B., Kuksenko, O., Dey, K. K., Jagadeesh, K. A., Thakore, P. I., Karayel, O., Yung, A. R., Rajagopalan, A., Meireles, A. M., Yang, K. D., Amir-Zilberstein, L., Delorey, T., Phillips, D., Raychowdhury, R., Moussion, C., Price, A. L., Hacohen, N., Doench, J. G., Uhler, C., Rozenblatt-Rosen, O., and Regev, A. Systematically characterizing the roles of e3-ligase family members in inflammatory responses with massively parallel perturb-seq. January 2023.

Hauser, A. and Bühlmann, P. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *J. Mach. Learn. Res.*, 13(1): 2409–2464, August 2012.

Immer, A., Schultheiss, C., Vogt, J. E., Schölkopf, B., Bühlmann, P., and Marx, A. On the identifiability and estimation of causal location-scale noise models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *ICML'23*, pp. 14316–14332. JMLR.org, July 2023.

Katz, D. A., Shanmugam, K., Squires, C., and Uhler, C. Size of interventional markov equivalence classes in random DAG models. *International Conference on Artificial Intelligence and Statistics*, 2019.

Lachapelle, S., Brouillard, P., Deleu, T., and Lacoste-Julien, S. Gradient-Based neural DAG learning. *International Conference on Learning Representations*, 2019.

Laumann, F., von Kügelgen, J., Park, J., Schölkopf, B., and Barahona, M. Kernel-based independence tests for causal structure learning on functional data. November 2023.

Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., and Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nat. Methods*, 15(12):1053–1058, December 2018.

Lopez, R., Hutter, J.-C., Pritchard, J., and Regev, A. Large-Scale differentiable causal discovery of factor graphs. *Adv. Neural Inf. Process. Syst.*, 2022.

Lorch, L., Krause, A., and Schölkopf, B. Causal modeling with stationary diffusions. October 2023.

M. Mooij, J. and Claassen, T. Constraint-Based causal discovery using partial ancestral graphs in the presence of cycles. In Peters, J. and Sontag, D. (eds.), *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pp. 1159–1168. PMLR, 2020.

Margaritis, D. and Thrun, S. Bayesian network induction via local neighborhoods. In Solla, S., Leen, T., and Müller, K. (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999.

Marx, A., Gretton, A., and Mooij, J. M. A weaker faithfulness assumption based on triple interactions. In de Campos, C. and Maathuis, M. H. (eds.), *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 451–460. PMLR, 2021.

Mokhtarian, E., Akbari, S., Ghassami, A., and Kiyavash, N. A recursive markov Boundary-Based approach to causal structure learning. *CD@KDD*, 2020.

Montagna, F., Mastakouri, A. A., Eulig, E., Noceti, N., Rosasco, L., Janzing, D., Aragam, B., and Locatello, F. Assumption violations in causal discovery and the robustness of score matching. *Neural Information Processing Systems*, 2023a.

Montagna, F., Noceti, N., Rosasco, L., Zhang, K., and Locatello, F. Scalable causal discovery with score matching. *Proceedings of Machine Learning Research*, April 2023b.

Montagna, F., Noceti, N., Rosasco, L., Zhang, K., and Locatello, F. Causal discovery with score matching on additive models with arbitrary noise. *Proceedings of Machine Learning Research*, April 2023c.

Mooij, J. M., Magliacane, S., and Claassen, T. Joint causal inference from multiple contexts. *J. Mach. Learn. Res.*, 21:1–108, March 2020.

Nazaret, A., Hong, J., Azizi, E., and Blei, D. Stable differentiable causal discovery. November 2023.

Pearl, J. An introduction to causal inference. *Int. J. Biostat.*, 6(2):Article 7, February 2010.

Pogodin, R., Deka, N., Li, Y., Sutherland, D. J., Veitch, V., and Gretton, A. Efficient conditionally invariant representation learning. *International Conference on Learning Representations*, 2022.

Polo, F. M., Sun, Y., and Banerjee, M. Conditional independence testing under misspecified inductive biases. In *Thirty-seventh Conference on Neural Information Processing Systems*, November 2023.

Raskutti, G. and Uhler, C. Learning directed acyclic graph models based on sparsest permutations. *Stat*, 7(1):e183, January 2018.

Regev, A., Teichmann, S. A., Lander, E. S., Amit, I., Benoist, C., Birney, E., Bodenmiller, B., Campbell, P., Carninci, P., Clatworthy, M., Clevers, H., Deplancke, B., Dunham, I., Eberwine, J., Eils, R., Enard, W., Farmer, A., Fugger, L., Göttgens, B., Hacohen, N., Haniffa, M., Hemberg, M., Kim, S., Klenerman, P., Kriegstein, A., Lein, E., Linnarsson, S., Lundberg, E., Lundeberg, J., Majumder, P., Marioni, J. C., Merad, M., Mhlanga, M., Nawijn, M., Netea, M., Nolan, G., Pe'er, D., Phillipakis, A., Ponting, C. P., Quake, S., Reik, W., Rozenblatt-Rosen, O., Sanes, J., Satija, R., Schumacher, T. N., Shalek, A., Shapiro, E., Sharma, P., Shin, J. W., Stegle, O., Stratton, M., Stubbington, M. J. T., Theis, F. J., Uhlen, M., van Oudenaarden, A., Wagner, A., Watt, F., Weissman, J., Wold, B., Xavier, R., Yosef, N., and Human Cell Atlas Meeting Participants. The human cell atlas. *Elife*, 6:e27041, December 2017.

Reisach, A. G., Seiler, C., and Weichwald, S. Beware of the simulated DAG! causal discovery benchmarks may be easy to game. *Adv. Neural Inf. Process. Syst.*, 2021.

Reizinger, P., Sharma, Y., Bethge, M., Schölkopf, B., Huszár, F., and Brendel, W. Jacobian-based causal discovery with nonlinear ICA. *Transactions on Machine Learning Research*, December 2022.

Replogle, J. M., Saunders, R. A., Pogson, A. N., Hussmann, J. A., Lenail, A., Guna, A., Mascibroda, L., Wagner, E. J., Adelman, K., Lithwick-Yanai, G., Iremadze, N., Oberstrass, F., Lipson, D., Bonnar, J. L., Jost, M., Norman, T. M., and Weissman, J. S. Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq. *Cell*, 185(14):2559–2575.e28, July 2022.

Richardson, T. A discovery algorithm for directed cyclic graphs. In *Proceedings of the Twelfth international conference on Uncertainty in artificial intelligence*, UAI'96, pp. 454–461, San Francisco, CA, USA, August 1996. Morgan Kaufmann Publishers Inc.

Rolland, P., Cevher, V., Kleindessner, M., Russell, C., Janzing, D., Schölkopf, B., and Locatello, F. Score matching enables causal discovery of nonlinear additive noise models. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18741–18753. PMLR, 2022.

Runge, J. Conditional independence testing based on a nearest-neighbor estimator of conditional mutual information. In Storkey, A. and Perez-Cruz, F. (eds.), *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 938–947. PMLR, 2018.

Schmidt, M., Niculescu-Mizil, A., and Murphy, K. Learning graphical model structure using l1-regularization paths. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2*, AAAI'07, pp. 1278–1283. AAAI Press, July 2007.

Sen, R., Suresh, A., Shanmugam, K., Dimakis, A., and Shakkottai, S. Model-Powered conditional independence test. *Adv. Neural Inf. Process. Syst.*, 2017.

Sethuraman, M. G., Lopez, R., Mohan, R., Fekri, F., Biancalani, T., and Hütter, J.-C. NODAGS-Flow: Nonlinear cyclic causal structure learning. *International Conference on Artificial Intelligence and Statistics*, 206, 2023.

Shah, R. D. and Peters, J. The hardness of conditional independence testing and the generalised covariance measure. *Ann. Stat.*, 48(3):1514–1538, June 2020.

Shi, C., Veitch, V., and Blei, D. Invariant representation learning for treatment effect estimation. *Uncertain. Artif. Intell.*, 2020.

Shimizu, S. and Hoyer, P. O. A linear non-gaussian acyclic model for causal discovery. https://jmlr.org/papers/volume7/shimizu06a/shimizu06a.pdf, 2006. Accessed: 2023-11-29.

Shimizu, S., Inazumi, T., Sogawa, Y., Hyvärinen, A., Kawahara, Y., Washio, T., Hoyer, P., and Bollen, K. DirectLiNGAM: A direct method for learning a linear Non-Gaussian structural equation model. *J. Mach. Learn. Res.*, 2011.

Spirtes, P. An anytime algorithm for causal inference. In Richardson, T. S. and Jaakkola, T. S. (eds.), *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*, volume R3 of *Proceedings of Machine Learning Research*, pp. 278–285. PMLR, 2001.

Spirtes, P. and Zhang, J. A uniformly consistent estimator of causal effects under the k-Triangle-Faithfulness assumption. *SSO Schweiz. Monatsschr. Zahnheilkd.*, 29 (4):662–678, November 2014.

Spirtes, P., Glymour, C., and Scheines, R. *Causation, Prediction, and Search*. Springer New York, 1993.

Strobl, E. V., Zhang, K., and Visweswaran, S. Approximate Kernel-Based conditional independence tests for fast Non-Parametric causal discovery. *Journal of Causal Inference*, 7(1), March 2019.

Triantafillou, S. and Tsamardinos, I. Constraint-based causal discovery from multiple interventions over overlapping variable sets. *J. Mach. Learn. Res.*, 16(66):2147–2205, 2015.

Tsamardinos, I., Brown, L. E., and Aliferis, C. F. The max-min hill-climbing bayesian network structure learning algorithm. *Mach. Learn.*, 65(1):31–78, October 2006.

Uhler, C., Raskutti, G., Buhlmann, P., and Yu, B. Geometry of the faithfulness assumption in causal inference. *Ann. Stat.*, 2012.

van de Geer, S. and Bühlmann, P. ℓ_0-penalized maximum likelihood for sparse directed acyclic graphs. *Ann. Stat.*, 41(2):536–567, 2013.

Van Hout, C. V., Tachmazidou, I., Backman, J. D., Hoffman, J. D., Liu, D., Pandey, A. K., Gonzaga-Jauregui, C., Khalid, S., Ye, B., Banerjee, N., Li, A. H., O'Dushlaine, C., Marcketta, A., Staples, J., Schurmann, C., Hawes, A., Maxwell, E., Barnard, L., Lopez, A., Penn, J., Habegger, L., Blumenfeld, A. L., Bai, X., O'Keeffe, S., Yadav, A., Praveen, K., Jones, M., Salerno, W. J., Chung, W. K., Surakka, I., Willer, C. J., Hveem, K., Leader, J. B., Carey, D. J., Ledbetter, D. H., Geisinger-Regeneron DiscovEHR

Collaboration, Cardon, L., Yancopoulos, G. D., Econo-
mides, A., Coppola, G., Shuldiner, A. R., Balasubrama-
nian, S., Cantor, M., Regeneron Genetics Center, Nelson,
M. R., Whittaker, J., Reid, J. G., Marchini, J., Overton,
J. D., Scott, R. A., Abecasis, G. R., Yerges-Armstrong,
L., and Baras, A. Exome sequencing and characterization
of 49,960 individuals in the UK biobank. *Nature*, 586
(7831):749–756, October 2020.

Verma, T. S. and Pearl, J. Equivalence and synthesis of
causal models. In *Probabilistic and Causal Inference*, pp.
221–236. ACM, New York, NY, USA, February 2022.

Vowels, M. J., Camgoz, N. C., and Bowden, R. D'ya like
DAGs? a survey on structure learning and causal discov-
ery. *ACM Comput. Surv.*, 55(4):1–36, November 2022.

Wei, D., Gao, T., and Yu, Y. DAGs with no fears: A closer
look at continuous optimization for learning bayesian
networks. *Adv. Neural Inf. Process. Syst.*, 2020.

Zhang, K., Peters, J., Janzing, D., and Schölkopf, B.
Kernel-based conditional independence test and appli-
cation in causal discovery. In *Proceedings of the Twenty-
Seventh Conference on Uncertainty in Artificial Intelli-
gence*, UAI'11, pp. 804–813, Arlington, Virginia, USA,
July 2011. AUAI Press.

Zhang, L. and Janson, L. Floodgate: inference for model-
free variable importance. July 2020.

Zheng, X., Aragam, B., Ravikumar, P., and Xing, E. DAGs
with NO TEARS: Continuous optimization for structure
learning. *Adv. Neural Inf. Process. Syst.*, 2018.

Zheng, X., Dan, C., Aragam, B., Ravikumar, P., and Xing,
E. P. Learning sparse nonparametric DAGs. *International
Conference on Artificial Intelligence and Statistics*, 108,
2020.

A. Review of graph learning methods

We attempt to learn graphs with no unobserved latent variables or cycles. There is an extensive history of algorithms to learn such graphs from data (Vowels et al., 2022). Methods can learn a graph by testing for conditional independence (Spirtes et al., 1993; Spirtes, 2001; Margaritis & Thrun, 1999), optimizing an objective (Chickering, 2002; van de Geer & Bühlmann, 2013; Raskutti & Uhler, 2018), performing independent component analysis (Shimizu & Hoyer, 2006; Shimizu et al., 2011; Reizinger et al., 2022), looking for nonlinearities (Rolland et al., 2022; Montagna et al., 2023b;c), and more (Immer et al., 2023; Gao et al., 2020; Reisach et al., 2021). Unfortunately, Nazaret et al. (2023) showed that almost all methods made strong assumptions on the form of the cause-effect relationships between variables or can not scale to more than 100 variables. Our work aims to perform flexible inference at large scale.

Recently a class of optimization-based graph learning methods were able to scale to learn from data of large complex systems by searching through the space of graphs at the same time as training flexible neural networks to model causal relationships (Zheng et al., 2018; Lachapelle et al., 2019; Zheng et al., 2020; Bello et al., 2022; Nazaret et al., 2023). These methods can also learn from data with interventions (Brouillard et al., 2020; Nazaret et al., 2023). Unfortunately, the model search can involve an unstable optimization problem over the enormous space of all graphs and has been shown to be unreliable in some settings (Wei et al., 2020; Nazaret et al., 2023; Deng et al., 2023). Nazaret et al. (2023) recently substantially improved the accuracy of these methods by first learning the moral graph to shrink the model search space. "Hybridizing" model search with some conditional independence testing is also the strategy of the most accurate classical graph learning methods (Tsamardinos et al., 2006; Bühlmann et al., 2014). Our work aims to further shrink the model search space, or learn the graph entirely, by testing for conditional independence at scale.

B. Details of the method

B.1. Orienting the edges of the skeleton

Once we have learned the skeleton, we finally need to decide the direction of its arrows. The equivalence class of a graph is determined by its skeleton and v-structures – variables X^n, X^k, X^m such that X^n and X^m are not adjacent and $X^n \to X^k \leftarrow X^m$ in G (Verma & Pearl, 2022). For all v-structures X^n, X^k, X^m we have that X^n and X^m are spouses, so they are adjacent in the moral graph but not the skeleton. As well, if X^n and X^m are spouses and $X^n - X^k - X^m$ in the skeleton, then if p is faithful to G, $X^k \notin \mathrm{SepSet}(X^n, X^m)$ if and only if $X^n \to X^k \leftarrow X^m$ in G.

To infer v-structures, we first pick a threshold $\eta_2 > 0$ and look for any triplet in the inferred skeleton $X^n - X^k - X^m$ such that X^n and X^m are adjacent in the inferred moral graph but not the inferred skeleton. We then decide if $X^k \in \mathrm{SepSet}(X^n, X^m)$ using the two learned parameters ψ^n and ψ^m from the two tests between X^n and X^m. We label the triplet a v-structure if $(\psi_k^n)^2 + (\psi_k^m)^2 < \eta_2$, that is if k is not in the separating set for either of the tests we did for the pair X^n, X^m, otherwise we label it not a v-structure. After we have labelled all of the v-structures, we can apply Meek's rules to orient many of the remaining edges as in the PC algorithm (Spirtes et al., 1993). We then use some heuristics described in Appendix B.3 to extract a single graph from the equivalence class.

B.2. Learning the graph with interventions with known targets

For clarity, assume p is faithful in this section. We first learn the moral graph over all variables $X^1, \ldots, X^N, I^1, \ldots, I^K$ with a minor modification. Predicting $\mathrm{MB}(I^k)$ is unreliable in practice so we do not use it to build the moral graph. Instead we just connect X^n and I^k in the moral graph if I^k is predicted to be in $\mathrm{MB}(X^n)$ – we pick a threshold $\eta_4 > 0$ and connect the two variables if $\alpha_{X^n, I^k} > \eta_4$.

Next we learn the skeleton over the variables $X^1, \ldots, X^N, I^1, \ldots, I^K$. We reduce the number of tests we perform with three techniques:

1) We do not test adjacencies of intervention variables I^k.

2) We learn the parents of intervened variables during the moral graph learning step. If X^n is adjacent to an intervention I^k with target $X^m \neq X^n$ in the moral graph then, since X^n cannot be the parent or child of I^k it must be its spouse – $X^n \in \mathrm{Pa}_G(X^m)$. Thus if X^n is adjacent to I^k in the moral graph then we label X^n a parent of X^m. We do not need to test the adjacency of X^n and X^m. [2]

---

[2] In principle we could orient every edge connected to the target of an intervention. Say we determine that X^n is adjacent to X^m in

3) We use the direction of edges learned in the moral graph step to shrink the search space of the adjacency test. The second statement in Prop. E.2 states we can always include parents of X^n and we can always exclude sinks – variables with no children – in the separating set. Thus, for $X^k \in \text{MB}(X^n)$, if X^k is a parent of X^n we fix $\psi_k^n = 1$; and if we have determined that every node that X^k is adjacent to in the moral graph is its parent then we fix $\psi_k^n = 0$.

Once we have learned the skeleton of the variables $X^1, \ldots, X^N$ and oriented some edges, we orient the remaining edges in $X^1, \ldots, X^N$ just as in Section B.1.

In Table 1 we see that using our method above allows DAT-Graph to learn skeletons more accurately in practice than by simply including the intervention variables as nodes in the graph (Naive JCI).

Table 1. **DAT-Graph accurately learns skeletons using intervention data.** Accuracy of predicting adjacencies in the experiment in Fig. 5 with $N = 200$ and 50% of variables intervened on.

MODEL	ERRORS (SKELETON SHD)
DAT-GRAPH	**51±9**
NAIVE JCI	69±9

B.3. Getting a graph from an equivalence class

After we are done applying Meek's rules, there may be a small number of non-oriented edges – we may have only identified the graph up to an equivalence class. To get a single graph, we iteratively randomly orient a randomly chosen unoriented edge and re-apply Meek's rules until all edges are oriented.

There may also be cycles in the graph we learn, $\hat{G}$, due to disagreeing tests. There are sophisticated methods to learn a graph in the case that tests disagree (Triantafillou & Tsamardinos, 2015). However, in our experiments there are usually only a small number of cycles in $\hat{G}$, so we take a simple approach to remove cycles. First we calculate the matrix $(\hat{G} + I)^N - I$. The n-th entry on the diagonal of this matrix is 0 if and only if the n-th node is not in a cycle (Zheng et al., 2018). While there are non-zero entries on the diagonal, we pick the node n that maximizes the value $((\hat{G} + I)^N - I)_{n,n}$ and remove an edge connected to this node that maximally reduces $\text{trace}((\hat{G} - I)^N - I)$.

B.4. Hyperparameters

We have four threshold hyperparameters: one for deciding the edges of the moral graph η_3, one for deciding edges in the skeleton η_1, one for deciding v-structures η_2, and one for deciding edges in the moral graph connected to intervention variables η_4. We choose $\eta_3 = 8 \times 10^{-3}$, $\eta_1 = 10^{-4}$, $\eta_2 = 0.2$, $\eta_4 = 10^{-3}$. Theorem 4.3 suggests that η_2 should be a larger number. We noticed however that a smaller value of η_2 resulted in more accurate graph recovery. We discuss why this might be in Appendix. E.4.2.

Each variable selection problem in inferring the moral graph has a sparsity parameter λ_n. We noticed when that $\sigma_n^2 = E[X^n - g_{\theta_n}(X^m)_{m \neq n}]^2$ could vary drastically from node to node. This caused the influence of the sparsity penalty to vary from node to node, making it challenging to get accurate graph recovery with a single threshold η_3. To address this issue, we found the scaling the sparsity parameter λ_n by σ_n^2 improved recovery of the moral graph. Thus for the n-th variable we use a sparsity penalty of $0.01 \times \sigma_n^2$ where σ_n^2 is estimated from the current minibatch.

We used batch sizes of size 256 in all cases. To train the neural networks to predict the moral graph, we used the Adam optimizer with parameters $\beta_1, \beta_2 = 0.9, 0.999$ and learning rate 10^{-4} and trained for 30000 minibatches. We train the models for all nodes in parallel on a GPU.

To train the networks to predict the skeleton we trained for 10000 minibatches and took alternating steps to update $\{\theta_1, \theta_2\}$ and ψ. For $\{\theta_1, \theta_2\}$ we used the Adam optimizer with parameters $\beta_1, \beta_2 = 0.9, 0.999$ and learning rate 3×10^{-4} while for ψ we used $\beta_1, \beta_2 = 0.9, 0.9$ and a learning rate of 3×10^{-4}. We train models for all tests in parallel on a GPU.

To predict the moral graph, we used 3 layer neural networks with 200 hidden units. We used a ReLU activation and included

---

G where X^m is the target of an intervention I^k. If I^k is not adjacent to X^n in the moral graph then X^n must be the child of X^m. In practice however, we have seen that this strategy is unreliable in the absence of a large amount of intervention data.

dropout and batchnorm between layers. We used a dropout probability of 0.1 between the first and second layer and a probability of 0.5 between the second and third. To predict the skeleton, we used 3 layer neural networks with 100 hidden units. We again used a ReLU activation and included dropout and batchnorm between layers.

B.5. Other details

Before learning the graph we normalize all variables in the data to have mean 0 and standard deviation 1.

We parameterize $\psi_n \in [0,1]$ as $\psi_n = \mathrm{sigmoid}(\gamma_n)$ for $\gamma_n \in (0, \infty)$. γ_n is the parameter we optimize by gradient descent.

When testing the adjacency between variables X^n, X^m, we have the choice of using either $E\mathrm{VarExplained}(X^n; X^m | \mathrm{MB}(X^n))$ or $E\mathrm{VarExplained}(X^m; X^n | \mathrm{MB}(X^n))$ as our measure for conditional independence. We use the later in experiments as we found it to make more accurate decisions.

We use combined samples from 5000 mini-batches to calculate $\hat{L}_{\mathrm{DAT}}(\psi)$.

C. Experimental details

C.1. Data simulation details

We simulate observational data just as in Nazaret et al. (2023). We use code from https://github.com/azizilab/sdcd under an MIT licence. Briefly, we generate a random undirected graph $\tilde{G}$ and a random permutation of $\{1, \ldots, N\}$, π. Then we have an edge $n \to m$ in the graph G if $\pi(n) > \pi(m)$ and n and m are connected in $\tilde{G}$. Next we model the functions

$$X^n \sim \tilde{h}_n \left(\frac{X^m - E[X^m]}{\mathrm{Std}(X^m)} \right)_{m \in \mathrm{Pa}_G(n)} + N(0,1)$$

where $\tilde{h}_n$ is a randomly initialized two layer neural network with ReLU activations and 100 hidden units. If $\mathrm{Pa}_G(X^n) = \emptyset$, we set $\tilde{h}_n = 0$. When simulating data with linear relations, we replace $\tilde{h}_n$ with a linear model with weights drawn from $N(0,1)$.

In our experiments with interventions, we assume that only a single variable is intervened on in each data point. If a variable X^n is intervened on then $X^n \sim 0.1 * N(0,1)$.

The SHD between two undirected graphs G_1, G_2 is the sum of the number of edges in G_1 but not G_2 and the number of edges in G_2 but not G_1. The SHD between two directed graphs is the sum of the SHD of their skeletons plus the number of edges pointed in the wrong direction.

We simulated Erdős-Renyi and scale free random graphs using code from Montagna et al. (2023a).

C.2. Ablation experiments

We performed ablations that demonstrate the benefits of our modelling decisions. We generated observational data as in Section 7 with $N = 200$ and $s = 4$; results are shown in Table 2 with mean skeleton SHD (defined in Section 7) and standard deviation across 3 replicates.

To demonstrate that our choice of representation provides a more accurate answer to the separating set selection problem we performed ablations where we replaced r_ψ with a 3 layer neural network $r_\psi : \mathbb{R}^M \to \mathbb{R}$ with 200 hidden units (**Neural net r_ψ**). We used a ReLU activation and included dropout and batchnorm between layers. We used a dropout probability of 0.1 between the first and second layer and a probability of 0.5 between the second and third. We then optimize ψ, the parameters of the neural network. We also performed ablations where we used noise distributions $(f_m)_m$ with thin tails – f_m were Gaussian densities – rather than the thick tailed distribution described in section 7 (**Gaussian $(f_m)_m$**).

To demonstrate that testing twice as discussed in Section 5.2 makes our method more reliable, we perform an ablation where we decide if X is adjacent to Y by randomly testing one of $X \perp\!\!\!\perp Y \mid U$ for some $U \subset \mathrm{MB}(X) \setminus \{Y\}$ or $X \perp\!\!\!\perp Y \mid U$ for some $U \subset \mathrm{MB}(Y) \setminus \{X\}$ (**Only one test**). We halve η_1 for this ablation.

In addition to the results shown in Table 2 we also demonstrate that solving the separating set selection problem is informative for learning the graph: we perform ablations where we do not learn the parameters ψ: we either test if two nodes are adjacent in the skeleton by testing their marginal independence ($\psi = 0$) or their conditional independence ($\psi = 1$). We show the

results in Table 2.

Table 2. **Ablations justify the choices in DAT-Graph in practice.** Accuracy of predicting adjacencies in a graph with $N = 200$ and 800 edges with standard deviations across three replicates. Details in Appendix C.2.

MODEL	ERRORS (SKELETON SHD)
DAT-GRAPH	**81±9**
NEURAL NET r_ψ	173±11
GAUSSIAN $(f_m)_m$	98±12
ONLY ONE TEST	177±16
TEST MARGINAL $\psi = 0$	125±19
TEST CONDITIONAL $\psi = 1$	308±14

C.3. Baseline methods

We implemented SDCD using the code from `https://github.com/azizilab/sdcd`. We used the hyperparameters described in Nazaret et al. (2023). In Nazaret et al. (2023) SDCD was trained on 2000 epochs on 10000 datapoints. When we added intervention data or trained on real RNA sequencing data, we scaled the number of epochs with the dataset size proportionally. We also scaled the parameter `gamma_increment` which controls the increment of the acyclicity penalty per epoch.

We also compare to a number of less scalable and flexible methods at small scale in Section D.2. **PC (Spirtes et al., 1993):** a classical method to learn the graph by looking for conditional independence relations; we implement this algorithm using a kernel test for conditional independence. We used code from `https://pywhy.org/dodiscover/` under an MIT licence with kernel threshold 0.05. **GES (Chickering, 2002):** a classical model selection procedure that performs the model search with greedy perturbations to the graph; it assumes all variables are jointly Gaussian. We implemented this using the code from Hauser & Bühlmann (2012). **CAM (Bühlmann et al., 2014):** a classical hybrid graph learning method that assumes that causal interactions are additive. We used code from `https://pywhy.org/dodiscover/` under an MIT licence. **NoGAM (Montagna et al., 2023c):** a recent method that learns conditional independence relations and infers a graph by assuming additive noise; it recently performed best in an array of small scale settings against other small scale graph learning methods (Montagna et al., 2023a). We used code from `https://pywhy.org/dodiscover/` under an MIT licence.

C.4. RNA sequencing data

We learned on single cell RNA sequencing data from a study of immunotherapy resistant cancer Frangieh et al. (2021). These cells had various genes perturbed by CRISPR knockdowns. We preprocessed this data as in Lopez et al. (2022) using code from `https://github.com/Genentech/dcdfg` under an Apache-2.0 licence. The preprocessed dataset included between 57523 and 87436 cells and measurements of $N = 1000$ genes. Data include observational and intervention samples. We created a test set by selecting 20% of intervention targets and holding out samples of those interventions. The test set had between 6984 and 11993 samples.

With default settings, SDCD predicts the conditional variance of variables with a neural network. We noticed that on this dataset, SDCD makes worse-than-trivial predictions with this setting (unless hybridized with the skeleton learned using DAT-Graph). Thus we used the setting `model_variance_flavor='parameter'`. We also used `finetune=True` when training to get a valid likelihood on the test set.

We implemented the `MLPGaussialModel` from DCDFG using code from `https://github.com/Genentech/dcdfg` with hyperparameters that were optimized according to Lopez et al. (2022) on the data, that is, $m = 20$, $\lambda = 10^{-3}$, trace exponential penalty.

Both SDCD and DCDFG model the data as coming from a Gaussian additive model. A trivial prediction for such a model is that all variables are generated from iid Gaussians. The mean negative log likelihood of this trivial model on a test set can be

calculated as

$$\sum_{n=1}^{N} \frac{1}{2} \log(2\pi \mathrm{Var}_{\mathrm{train}}(X^n)) + \frac{1}{2} \frac{E_{\mathrm{test}}\left[X^n - E_{\mathrm{train}} X^n\right]^2}{\mathrm{Var}_{\mathrm{train}}(X^n)}.$$

To build the hybrid method, we reasoned that the same hyperparameters that are optimal for graph recovery as measured by SHD may not be optimal for intervention prediction. It may be the case for example that leaving out a causal arrow in the graph may harm prediction much more than including spuriously inferred edges. Thus we used the same hyperparameters as above but picked $\eta_3 \in \{0.001, 0.003, 0.005, 0.008\}$ based on what minimized the fit on the training data on the "control" dataset set. We found $\eta_3 = 0.001$ lead to the best fit on the training data and used this value for experiments in the main text.

D. Further experimental results
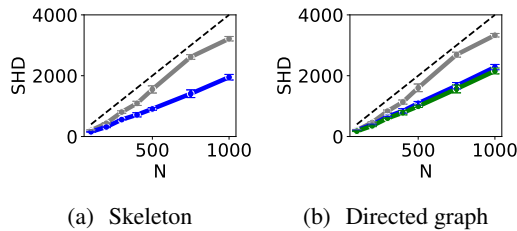
D.1. Appendix to main text figures

(a) Skeleton   (b) Directed graph

Figure 7. **DAT-Graph can accurately learn from data of a scale-free random graph.** We perform the experiment in Fig. 3 with scale-free random graphs. We plot the mean SHD and standard deviation across 3 replicates. The legend is the same as that of Fig. 3.
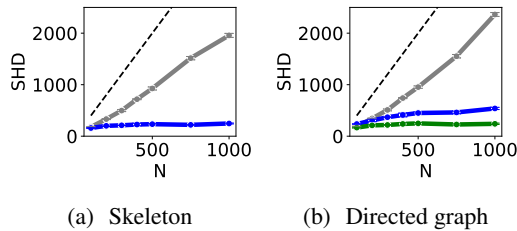

(a) Skeleton   (b) Directed graph

Figure 8. **DAT-Graph can accurately learn from data with linear relations.** We perform the experiment in Fig. 3 with linear relations between variable. We plot the mean SHD and standard deviation across 3 replicates. The legend is the same as that of Fig. 3.
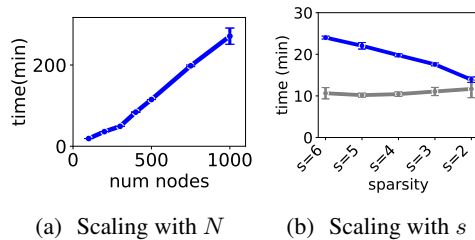

(a) Scaling with N (b) Scaling with s

Figure 9. **DAT-Graph scales linearly with N and is more efficient on sparser graphs.** A) For $s = 4$, we plot the wall time of DAT-Graph for various values of N. B) For $N = 100$ we plot the wall time of DAT-Graph for various values of s. Error bars are standard deviations over 3 replicates. The legend is the same as that of Fig. 3.
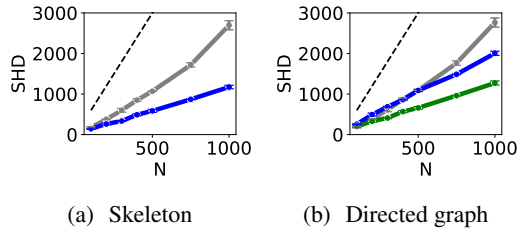
(a) Skeleton   (b) Directed graph

Figure 10. **DAT-Graph and the hybrid method learn large graphs accurately even when they are dense.** We perform the experiment in Fig. 3 with $s = 6$. We plot the mean SHD and standard deviation across 3 replicates. The legend is the same as that of Fig. 3.
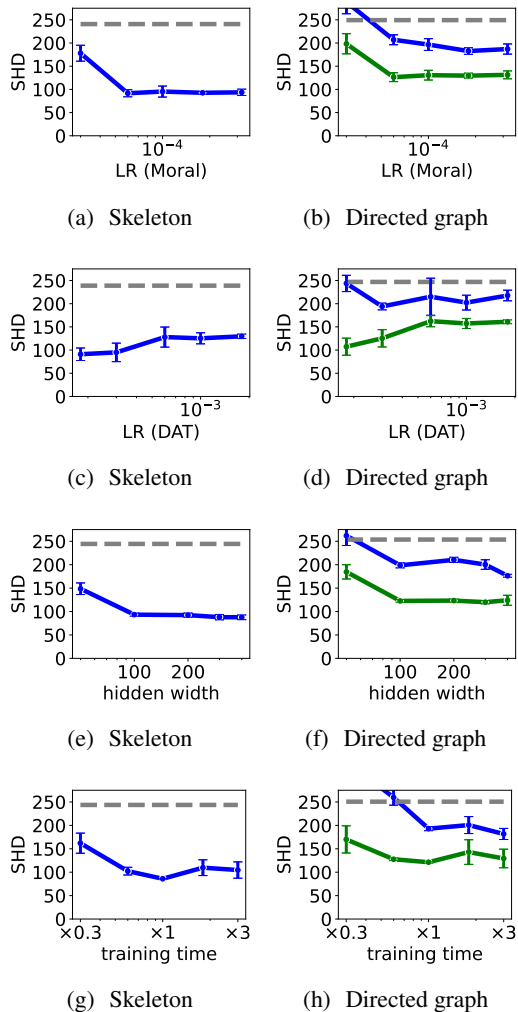

(a) Skeleton   (b) Directed graph

(c) Skeleton   (d) Directed graph

(e) Skeleton   (f) Directed graph

(g) Skeleton   (h) Directed graph

Figure 11. **DAT-Graph is robust to the choice of neural network hyperparameters.** We perform the experiment in Fig. 5 with $N = 200$, intervening on 50% of variables, varying the learning rate of the moral graph learning step, the DAT step, the hidden widths of the nerual networks, and the neural network training times over an order of magnitude. We plot the mean SHD and standard deviation across 3 replicates. The legend is the same as that of Fig. 3.
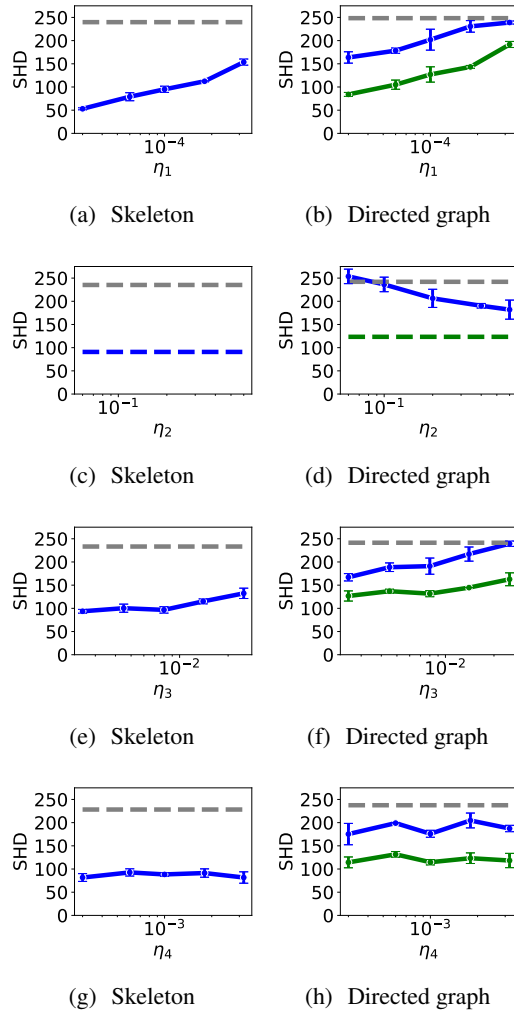
(a) Skeleton (b) Directed graph

(c) Skeleton (d) Directed graph

(e) Skeleton (f) Directed graph

(g) Skeleton (h) Directed graph

Figure 12. **DAT-Graph is robust to the choice of threshold hyperparameters.** We perform the experiment in Fig. 5 with $N = 200$, intervening on 50% of variables, varying the parameters $\eta_1, \eta_2, \eta_3, \eta_4$ over an order of magnitude. We plot the mean SHD and standard deviation across 3 replicates. The legend is the same as that of Fig. 3.
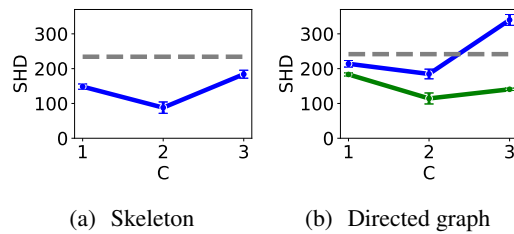

(a) Skeleton (b) Directed graph

Figure 13. **DAT-Graph best learns graphs when using two moments in its variance explained statistic.** We perform the experiment in Fig. 5 with $N = 200$ intervening on 50% of variables. We use DAT with the statistic $\sum_{c=1}^{C} E\mathbb{V}(T_c(X); Y | \tilde{Z}_\psi)$ where $T_c(x) = x^c$. We plot the mean SHD and standard deviation across 3 replicates. The legend is the same as that of Fig. 3.

20

D.2. Experiments with small N

Here we show that DAT-Graph can accurately learn graphs at small N. We compare to a classical testing approach with a flexible conditional independence test (PC), a classical explicit graph search procedure (GES), a classical hybrid method (CAM), and a modern method that make learns conditional independence relations by looking for non-linear interactions (NoGAM).

We trained all of these models on data with $N = 30$ and $s = 3$. We could not scale NoGAM to learn from 10000 datapoints in under 15 hours of wall time so we trained all models on 6000 datapoints. The exception is the PC algorithm, which due to the cost of the flexible conditional independence test, could not learn from 600 datapoints in under 15 hours of wall time so we trained this model on 300 datapoints.

In Table 3 we see that GES, CAM, and NoGAM perform worse than trivial. This could be because the complex interactions in our data violate the Gaussianity assumption of GES, and the additivity assumption of CAM. As well, since neural networks with ReLU activations are locally linear, the nonlinearity assumptions in NoGAM are also violated. The PC algorithm performs slightly better than trivial – its performance could be limited by its inability to scale to a larger training set size. On the other hand, DAT-Graph and the Hybrid method are able to learn the graph most accurately.

Table 3. **DAT-Graph can accurately learn a graph at low N.** Mean SHDs and standard deviation for inferred graphs across 3 replicates.

MODEL	SKELETON SHD	DIRECTED GRAPH SHD
NUMBER EDGES	60	60
PC (300 DATAPOINTS)	44±2	56±5
GES	99±85	104±23
CAM	79±33	87±18
NoGAM	79±24	85±14
SDCD	25± 4	26±4
DAT-GRAPH	**8±1**	16±4
HYBRID	-	**11± 3**

D.3. Alternative hybrid model for RNA sequencing data

To demonstrate that the benefit of hybridizing DAT-Graph and SDCD did not simply come from restricting the edges learned by SDCD during training, we also compare to another hybrid model – in **SDCD(With Graph)** we use the graph inferred by another SDCD model in place of the one learned by DAT-Graph. In table 4 we show that the performance of this hybrid model is almost identical to that of SDCD.

Table 4. **Improved prediction from using DAT-Graph does not come exclusively from training SDCD.** We log the learned mutual information for three datasets as in Fig. 6.

MODEL	CONTROL	IFN	CO-CULTURE
SDCD	5.2	12.7	4.3
SDCD(WG)	5.4	12.6	4.7
HYBRID	**26.7**	**28.3**	**20.6**

E. Theory

E.1. Proof of Prop. 4.4

Proposition E.1. *(Proof of Prop. 4.4) Even when restricted to the case where $X, Y, Z_1, \ldots, Z_M$ are jointly Gaussian with known non-singular covariance matrix, the separating set selection problem is NP-Hard.*

Proof. We will show that the subset sum problem, which is known to be NP-hard, reduces to the above problem. The subset sum problem is as follows: given a set of numbers $a_1, \ldots, a_M, T \in \mathbb{R}$, is there a subset $S \subset \{a_m\}_{m=1}^{M}$ such that

$\sum_{a \in S} a = T$?

Let $\epsilon_X, \epsilon_Y, Z_1, \ldots, Z_M$ be jointly independent Gaussian variables with variance one, let $X = \epsilon_X + \sum_{m=1}^{M} Z_m$ and $Y = \epsilon_Y - T\epsilon_X + \sum_{m=1}^{M} a_m Z_m$. Now if $S \subset \{Z_m\}_{m=1}^{M}$ then $\mathrm{Cov}(X, Y|S) = -T + \sum_{Z_m \notin S} a_m$. So if there is a $S \subset \{Z_m\}_{m=1}^{M}$ such that $X \perp\!\!\!\perp Y|S$ then $0 = \mathrm{Cov}(X, Y|S) = -T + \sum_{Z_m \notin S} a_m$ and $\sum_{Z_m \notin S} a_m = T$. Similarly if there is no such subset then the answer to the subset sum problem is negative. $\square$

E.2. Proof of Prop. 5.2

Proposition E.2. *(Proof of Prop. 5.2) Assume p is faithful. X^n and X^m are adjacent in G if and only $X^n \not\perp\!\!\!\perp X^m \mid U$ for any $U \subset \mathrm{MB}(X^n) \setminus \{X^m\}$. If $X^n \perp\!\!\!\perp X^m \mid U$ for some $U \subset \mathrm{MB}(X^n) \setminus \{X^m\}$ then $X^n \perp\!\!\!\perp X^m \mid U \cup \mathrm{Pa}_G(X^n) \setminus \{X^m \mid \mathrm{De}_G(X^m) = \emptyset\}$.*

Proof. (Adapted from Margaritis & Thrun (1999)) This result is obvious if X^n and X^m are adjacent or $X^m \notin \mathrm{MB}(X^n)$, so assume X^m is a spouse of X^n. If X^m is not a descendant of X^n, then if $U = \mathrm{Pa}_G(X^n)$ then $X^n \perp\!\!\!\perp X^m \mid U$. If X^m is a descendant of X^n then include in U $\mathrm{Pa}_G(X^n)$ and all variables in $\mathrm{Ch}_G(X^n)$ that are ancestors of X^m and the parents of these variables. Say we have a d-connecting path from X^n to X^m. Since we have conditioned on all parents of X^n the path must have an arrow out of X^n. Say the first edge is $X^n \to X^k$. By the definition of U, X^k cannot be an ancestor of X^m, so the path must eventually encounter its first collider at an $X^l \in U$. By the definition of U, X^l must be a child of X^n that is an ancestor of X^m or a parent of such a child. In either case, X^l is an ancestor of X^m. Thus, X^k is an ancestor of X^m, a contradiction. $\square$
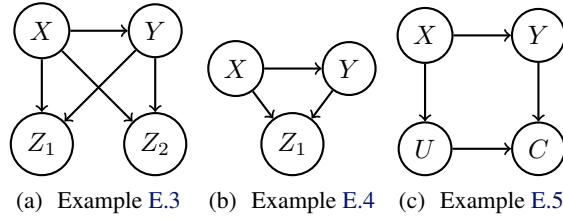
E.3. Counter-examples

(a) Example E.3   (b) Example E.4   (c) Example E.5

Figure 14. **Graphs considered in counterexamples.**

Example E.3. **(Proof of Example 4.2)** There are jointly Gaussian variables X, Y, Z_1, Z_2 that are faithful to some graph such that $X \not\perp\!\!\!\perp Y|\{Z_m\}_{m \in S}$ for any $S \subset \{1, 2\}$ but if $\{r_\psi\}_{\psi \in \Psi}$ is the space of linear functions, there is a ψ^* such that $X \perp\!\!\!\perp Y|r_{\psi^*}(Z_m)_{m=1}^{M}$.

Proof. Let $\epsilon_X, \epsilon_Y, \epsilon_{Z_1}, \epsilon_{Z_2} \sim N(0, 1)$ iid. Define $X = \epsilon_X$, $Y = X + \epsilon_Y$, $Z_1 = X + Y + \sqrt{\frac{1}{2}}\epsilon_{Z_1}$ and $Z_2 = X + Y + \sqrt{3 + \frac{1}{2}}\epsilon_{Z_2}$. The covariance matrix of $\{X, Y, Z_1, Z_2\}$ is

$$\begin{bmatrix} 1 & 1 & 2 & 2 \\ 1 & 2 & 3 & 3 \\ 2 & 3 & 5 + \frac{1}{2} & 5 \\ 2 & 3 & 5 & 8 + \frac{1}{2} \end{bmatrix}.$$

Then one can calculate that

$$\mathrm{Cov}(X, Y) = 1, \mathrm{Cov}(X, Y|Z_1) = 1 - \frac{6}{5 + \frac{1}{2}}, \mathrm{Cov}(X, Y|Z_2) = 1 - \frac{6}{8 + \frac{1}{2}}, \mathrm{Cov}(X, Y|Z_1, Z_2) = 1 - \frac{24}{21 + \frac{3}{8}}. \quad (5)$$

22

so $X \not\perp\!\!\!\perp Y|S$ for any $S \subset \{Z_1, Z_2\}$. Further calculation shows that this is faithful to the graph in Fig. 14(a). Now define $Z_3 = \frac{1}{2}(Z_1 + Z_2)$. The covariance matrix of $\{X, Y, Z_3\}$ is

$$\begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 6 \end{bmatrix}.$$

Thus $\text{Cov}(X, Y|Z_3) = 0$ so $X \perp\!\!\!\perp Y|Z_3$. □

Example E.4. There are jointly Gaussian variables X, Y, Z_1 that are faithful to some graph such that $X \not\perp\!\!\!\perp Y$ and $X \not\perp\!\!\!\perp Y \mid Z_1$ but if f_1 is a Gaussian density then there is a $\psi^* \in [0, 1]$ such that $X \perp\!\!\!\perp Y|\tilde{Z}_{1,\psi^*}$.

Proof. Let $\epsilon_X, \epsilon_Y, \epsilon_{Z_1} \sim N(0, 1)$ iid. Define $X = \epsilon_X, Y = X + \epsilon_Y, Z_1 = X + Y + \sqrt{\frac{1}{2}}\epsilon_{Z_1}$. The covariance matrix of $\{X, Y, Z_1\}$ is

$$\begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 5 + \frac{1}{2} \end{bmatrix}.$$

Then one can calculate that this distribution is faithful to the graph in Fig. 14(b). Now pick $\psi_1^* = 2 - \sqrt{2}$ and define $\tilde{Z}_1 = \psi_1^* Z_1 + (1 - \psi_1^*)N_1$ where N_1 is an independent standard normal. $\psi_1^*\epsilon_{Z_1} + (1 - \psi_1^*)N_1$ is a mean zero normal distribution of variance

$$(2 - \sqrt{2})^2\frac{1}{2} + (\sqrt{2} - 1)^2 = (2 - \sqrt{2})^2\frac{1}{2} + (2 - \sqrt{2})^2\frac{1}{2} = (2 - \sqrt{2})^2 = \psi_1^{*2}.$$

Thus $\tilde{Z}_1 = \psi_1^*(X + Y + U)$ for some independent standard normal U. The covariance matrix of $\{X, Y, \tilde{Z}_1\}$ is

$$\begin{bmatrix} 1 & 1 & 2\psi_1^* \\ 1 & 2 & 3\psi_1^* \\ 2\psi_1^* & 3\psi_1^* & 6\psi_1^{*2}. \end{bmatrix}.$$

Thus $\text{Cov}(X, Y|\tilde{Z}_1) = 0$ so $X \perp\!\!\!\perp Y|\tilde{Z}_1$. □

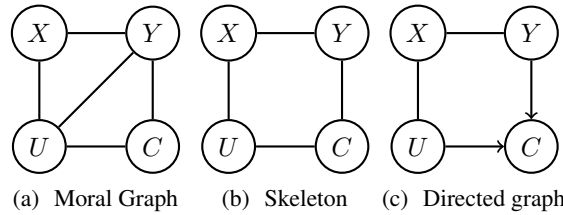(a) Moral Graph   (b) Skeleton   (c) Directed graph

Figure 15. **Stages of DAT-Graph in Example. 5.3**

Example E.5. (Proof of Example 5.3) There are a set of four jointly Gaussian random variables that are not faithful such that DAT-Graph recovers the correct graph.

Proof. Let $\epsilon_X, \epsilon_Y, \epsilon_U, \epsilon_C \sim N(0, 1)$ iid. Define $X = \epsilon_X, Y = X + \epsilon_Y, U = X + \epsilon_U, C = 2Y + U + \epsilon_C$. $\{X, Y, U, C\}$ have covariance matrix

$$\begin{bmatrix} 1 & 1 & 1 & 3 \\ 1 & 2 & 1 & 5 \\ 1 & 1 & 2 & 4 \\ 3 & 5 & 4 & 15 \end{bmatrix}.$$

$\{X, Y, U, C\}$ are generated according to the graph in Fig. 14(c). However, since $\text{Cov}(X, Y|C) = 0$ this distribution is not faithful to the graph. Note it is also not adjacency faithful (Uhler et al., 2012). When a distribution is not faithful it can correspond to multiple graph equivalence classes, so the notion of a "correct" graph is more delicate. We show that the

graph in Fig. 14(c) is correct in the sense that it is a sparsest Markov graph (Raskutti & Uhler, 2018). Then we show that DAT-Graph recovers this "correct" graph.

Say G is a graph that is Markov for this distribution, that is, it satisfies Eqn. 2 and therefore, for any disjoint $A, B, C \subset \{X, Y, U, C\}$ if A is d-separated from B in G by C, then $A \perp\!\!\!\perp B \mid C$. We have

$$\mathrm{Cov}(X, U) = 1, \mathrm{Cov}(X, U|Y) = \frac{1}{2}, \mathrm{Cov}(X, U|C) = \frac{3}{15}, \mathrm{Cov}(X, U|Y, C) = \frac{59}{55}$$

so X must be connected to U in G. Next we have

$$\mathrm{Cov}(Y, C) = 5, \mathrm{Cov}(Y, C|X) = 2, \mathrm{Cov}(Y, C|U) = 3, \mathrm{Cov}(Y, C|X, U) = 2$$

so Y must be connected to C in G. Finally we have

$$\mathrm{Cov}(U, C) = 4, \mathrm{Cov}(U, C|X) = 1, \mathrm{Cov}(U, C|Y) = \frac{3}{2}, \mathrm{Cov}(U, C|X, Y) = 1$$

so U must be connected to C in G. These three cannot be the only edges in G however as $\mathrm{Cov}(X, Y), \mathrm{Cov}(X, Y|U) \neq 0$. Thus the graph in Fig. 14(c) is a sparsest Markov graph.

We now show that DAT-Graph recovers the graph in Fig. 14(c). Some calculations show

$$\mathrm{MB}(X) = \{U, Y\}, \mathrm{MB}(Y) = \{X, C, U\}, \mathrm{MB}(U) = \{X, C, Y\}, \mathrm{MB}(C) = \{U, Y\}.$$

DAT-Graph joins nodes Z_1, Z_2 in the moral graph if $Z_1 \in \mathrm{MB}(Z_2)$ or $Z_2 \in \mathrm{MB}(Z_1)$, so we get the moral graph in Fig. 15(a).

Next we learn the skeleton. Since $\mathrm{Cov}(U, Y|X) = 0$, the edge between Y and U is removed with $\mathrm{SepSet}(Y, U) = \{X\}$; Note $\mathrm{Cov}(U, Y|X, C), \mathrm{Cov}(U, Y|C) \neq 0$ so C can not be in $\mathrm{SepSet}(Y, U)$. There is a subset of $\mathrm{MB}(Y)$ that makes X and Y conditionally independent, namely $\mathrm{Cov}(X, Y|C) = 0$. However, $\mathrm{Cov}(X, Y), \mathrm{Cov}(X, Y|U) \neq 0$ so there is no such subset of $\mathrm{MB}(X)$. DAT-Graph only removes an edge if both adjacency tests decide there is no edge. Therefore we learn the skeleton in Fig. 15(b).

Finally, DAT-Graph uses the removed edge between Y and U to find v-structures. Since $C \notin \mathrm{SepSet}(Y, U)$, we must have a v-structure $Y \to C \leftarrow U$. All other triplets are labelled non-v-structures. Thus we learn the graph equivalence class in Fig. 15(c). This is the equivalence class of the "correct" graph Fig. 15(b). $\square$

E.4. Proof of Theorem 4.3

In this section we will consider a set of random variables $X, Y, Z_1, \ldots, Z_M$ in $\mathbb{R}$. Calling $Z = (Z_m)_{m=1}^M$, we are interested in the separating set selection problem – evaluating if there is a "separating set" of variables $S \subset Z$ such that $X \perp\!\!\!\perp Y|S$. This is a challenging problem because it is discrete so, we relax it into the separating representation search. We define independent noise variables $N_1, \ldots, N_M$ and define noised versions of Z by $\tilde{Z}_{m, \psi_m} = \psi_m Z_m + (1 - \psi_m) N_m$ where $\psi_m \in [0, 1]$ are variables that control how much information about Z_m we obtain by observing $\tilde{Z}_{m, \psi_m}$. We now are interested in finding $\psi = (\psi_m)_m$ such that $X \perp\!\!\!\perp Y|\tilde{Z}_\psi$.

If S is a separating set then picking $\psi_m = \mathbb{1}(Z_m \in S)$ gives $X \perp\!\!\!\perp Y|\tilde{Z}_\psi$ so there is also a separating representation. However, the contrary might not be true as shown Example E.4 Here in Thm. E.7 we show that if $(f_m)_m$ have thick enough tails then if there is a separating representation there is also a separating set. Furthermore, if $X \perp\!\!\!\perp Y|\tilde{Z}_\psi$ then we can recover a separating set $\{Z_m\}_{\psi_m=1}$.

The idea of the proof is that if f_m have thick tails then the values of the noise N_m can be large. Then if we observe a large $\tilde{Z}_{m, \psi_m}$ all we can conclude is that N_m took a large value – we learn little about Z_m. Thus, conditioning on $(\tilde{Z}_{m, \psi_m})_m$ is similar to observing Z_m if $\psi_m = 1$ and not observing Z_m if $\tilde{Z}_{m, \psi_m}$ is large and $\psi_m < 1$. Thus, if $X \perp\!\!\!\perp Y|\{\tilde{Z}_{m, \psi_m}\}_m$ then $X \perp\!\!\!\perp Y|\{Z_m\}_{\psi_m=1}$.

Statement and verification of assumption tails of $f_1, \ldots, f_M$ We make the following assumption.

Assumption E.6. (f_m **has thicker tails than** $p(z)$) We assume $(f_m)_m$ are positive bounded symmetric functions on $\mathbb{R}$ that are decreasing and piece-wise differentiable on $(0, \infty)$.

- We assume the derivative of $\log f_m$ approaches 0 and for any value $z_m > 0$, $(\log f_m)'(z_m) \leq (\log f_m)'(z_m + 1)$.

- For any subset $S \subset \{1, \ldots, M\}$ and almost any set of values $(z_m)_{m \notin S}$ define the tail probabilities $p(L) = p(\|(z_m)_{m \in S}\|_\infty \geq L|(z_m)_{m \notin S})$. We assume that for any set of positive numbers, $(h_m)_{m \in S}$,

$$\sum_{L=0}^{\infty} \frac{p(L + R)}{\prod_{m=1}^{M} f_m(h_m(L + 2R + 1))} \to 0 \text{ as } R \to \infty. \tag{6}$$

The condition on the derivative of $\log f_m$ assumes that f_m is thicker than $f_m(z_m) \propto \exp(-|z_m|)$. Equation 6 assumes that the tail of $p(z)$ decreases faster than the tail of f_m. It essentially assumes that "f_m has thicker tails than $p(z)$".

Although its statement is technical, the assumption is easy to satisfy because A) we often have some idea of the tails of p, or can measure them, and B) we can pick f to have tails as thick as we would like.

We are often willing to assume that p is sub-Gaussian or sub-exponential; for example, RNA counts, based on knowledge of the biological generating process, are regularly assumed to come from a Poisson or negative binomial distribution, which are sub-exponential (Lopez et al., 2018). Below in Appendix E.4.1 we verify the assumption in the setting of our experiments in Section 7 using only the fact that p is sub-Gaussian; a similar argument can be used to verify the assumption in the case that p is sub-exponential. When one is not willing to make such an assumption, there are a number of methods to estimate the tails of a distribution. Once we have an idea of the thickness of p's tails, we can pick f to be as thick as necessary. We hypothesize that there is a tradeoff such that an f with thin tails may lead to separating representations when there are no separating sets, while an f with very thick tails may make training unstable, or reduce statistical efficiency; thus in our experiments in Section 7 we pick an f that is just thick enough to satisfy the assumption.

Statement and proof of theorem To construct our noised variables $\tilde{Z}$, we first pick our noise densities $f_1, \ldots, f_M$ and our noise parameters $\psi_1, \ldots, \psi_M \in [0, 1]$. Then we observe noised variables $\tilde{z}_m = \psi_m z_m + (1 - \psi_m) n_m$ where $n_m \sim f_m(n_m)dn_m$. Thus, if $\psi_m < 1$, $p(\tilde{z}_m|z_m) \propto f_m\left(\frac{1}{1-\psi_m}\tilde{z}_m - \frac{\psi_m}{1-\psi_m}z_m\right)$. Thus we get a posterior

$$p((z_m)_m|(\tilde{z}_m)_m, (\psi_m)_m, (f_m)_m) \propto \prod_{m \,|\, \psi_m \neq 1} f_m\left(\frac{1}{1-\psi_m}\tilde{z}_m - \frac{\psi_m}{1-\psi_m}z_m\right) dp((z_m)_{\psi_m < 1}|(z_m)_{\psi_m = 1}).$$

Now we can prove the theorem.

Theorem E.7. *Assume Assumption E.6. If there is a $(\psi_m)_m$ such that $X \perp\!\!\!\perp Y|\tilde{Z}_\psi$ then $X \perp\!\!\!\perp Y|\{Z_m\}_{\psi_m=1}$.*

Proof. Assume $X \perp\!\!\!\perp Y|\tilde{Z}_\psi$. We will show that as $(\tilde{z}_m)_{\psi_m < 1}$ get large, the posterior converges to the marginal in total variation

$$p((z_m)_m|(\tilde{z}_m)_m, (\psi_m)_m, (f_m)_m) \to p((z_m)_{\psi_m < 1}|(z_m)_{\psi_m = 1}).$$

Then for any measurable sets A, B we have

$$\begin{aligned}0 =\,& p(X \in A, Y \in B|(\tilde{z}_m)_m, (\psi_m)_m, (f_m)_m) - p(X \in A|(\tilde{z}_m)_m, (\psi_m)_m, (f_m)_m)p(X \in A|(\tilde{z}_m)_m, (\psi_m)_m, (f_m)_m) \\ \to\,& p(X \in A, Y \in B|(z_m)_{\psi_m=1}) - p(X \in A|(z_m)_{\psi_m=1})p(X \in A|(z_m)_{\psi_m=1}).\end{aligned}$$

This proves that $X \perp\!\!\!\perp Y \mid \{Z_m\}_{\psi_m=1}$.

We fix the values $(z_m)_{\psi_m=1}$ drop the dependence on these conditioned variables. We also drop the dependence on ψ_m by redefining $f_m(z_m) = f_m(\frac{\psi_m}{1-\psi_m}z_m)$ and $a_m = \frac{1}{1-\psi_m}\tilde{z}_m$. We set all a_m equal to a single large integer a. Thus we write the posterior

$$p(z|a) \propto \prod_{m-1}^{M} f_m(z_m - a)dp(z).$$

Now we show that this posterior converges in total variation to the marginal $p(z)$ as $a \to \infty$. For any $R > 0$,

$$
\int \left| \frac{\prod_{m=1}^{M} f_m(z_m - a)}{\int \prod_{m=1}^{M} f_m(z_m - a) dp(z)} - 1 \right| dp(z) = \int_{\|z\|_\infty \leq R} \left| \frac{\prod_{m=1}^{M} f_m(z_m - a_m)}{\int_{\|z\|_\infty \leq R} \prod_{m=1}^{M} f_m(z_m - a) dp(z)} - 1 \right| dp(z)
$$

$$
+ \int_{\|z\|_\infty \leq R} \left| \frac{\prod_{m=1}^{M} f_m(z_m - a)}{\int_{\|z\|_\infty \leq R} \prod_{m=1}^{M} f_m(z_m - a) dp(z)} \right.
$$

$$
\left. - \frac{\prod_{m=1}^{M} f_m(z_m - a)}{\int \prod_{m=1}^{M} f_m(z_m - a) dp(z)} \right| dp(z) \tag{7}
$$

$$
+ \int_{\|z\|_\infty > R} \left| \frac{\prod_{m=1}^{M} f_m(z_m - a)}{\int \prod_{m=1}^{M} f_m(z_m - a) dp(z)} \right| dp(z)
$$

$$
= \int_{\|z\|_\infty \leq R} \left| \frac{\prod_{m=1}^{M} f_m(z_m - a)}{\int_{\|z\|_\infty \leq R} \prod_{m=1}^{M} f_m(z_m - a) dp(z)} - 1 \right| dp(z)
$$

$$
+ 2 \frac{\int_{\|z\|_\infty > R} \prod_{m=1}^{M} f_m(z_m - a) dp(z)}{\int \prod_{m=1}^{M} f_m(z_m - a) dp(z)}.
$$

We now show both of these terms vanish as $R, a \to \infty$.

For the first term, note for all $\|z\|_\infty \leq R$.

$$
\prod_{m=1}^{M} \frac{f_m(a + R)}{f_m(a - R)} \leq \frac{\prod_{m=1}^{M} f_m(z_m - a)}{\int_{\|z\|_\infty \leq R} \prod_{m=1}^{M} f_m(z_m - a) dp(z)} \leq \frac{1}{p(\|z\|_\infty \leq R)} \prod_{m=1}^{M} \frac{f_m(a - R)}{f_m(a + R)}
$$

By our assumption on the derivatives of $\log f_m$,

$$
\frac{f_m(a - R)}{f_m(a + R)} = \exp \left(\log f_m(a - R) - \log f_m(a + R) \right)
$$

$$
= \exp \left(- \int_{a-R}^{a+R} (\log f_m)'(s) ds \right).
$$

This quantity $\to 1$ if $R \to \infty$ slowly enough and $a \to \infty$. Thus, if $R \to \infty$ slowly enough and $a \to \infty$, the first term in Eqn. 7 is approaches

$$
\int_{\|z\|_\infty \leq R} \left| \frac{1}{p(\|z\|_\infty \leq R)} - 1 \right| dp(z) = p(\|z\|_\infty \geq R) \to 0.
$$

For the second term, first note that if $L_2 > L_1$ then

$$
\frac{f_m(L_1)}{f_m(L_2)} = \exp \left(\log f_m(L_1) - \log f_m(L_2) \right)
$$

$$
= \exp \left(- \int_{L_1}^{L_2} (\log f_m)'(s) ds \right)
$$

$$
\leq \exp \left(- \int_0^{L_2 - L_1} (\log f_m)'(s) ds \right)
$$

$$
= \frac{f_m(0)}{f_m(L_2 - L_1)}.
$$

Assume we have picked R large enough that $p(\|z\|_\infty \leq R) \geq 1/2$. Define the annulus probabilities $\tilde{p}(L) = p(L + 1 >$

$\|(z_m)_{m\in S}\|_\infty \geq L)$ The second term can be broken up into annuluses

$$\frac{\int_{\|z\|_\infty > R} \prod_{m=1}^M f_m(z_m - a) dp(z)}{\int \prod_{m=1}^M f_m(z_m - a) dp(z)} \leq \sum_{L=R}^\infty \frac{\int_{L \leq \|z\|_\infty < L+1} \prod_{m=1}^M f_m(z_m - a) dp(z)}{\int_{\|z\|_\infty \leq R} \prod_{m=1}^M f_m(z_m - a) dp(z)}$$

$$\leq \sum_{L=R}^{a-1} \frac{p(L \leq \|z\|_\infty < L+1) \prod_{m=1}^M f_m(a - (L+1))}{p(\|z\|_\infty \leq R) \prod_{m=1}^M f_m(a + R)}$$

$$+ \sum_{L=a}^\infty \frac{p(L \leq \|z\|_\infty < L+1) \prod_{m=1}^M f_m(L - a)}{p(\|z\|_\infty \leq R) \prod_{m=1}^M f_m(a + R)}$$

$$\leq 2 p(\|z\|_\infty \leq R)^{-1} \sum_{L=R}^{a-1} \tilde{p}(L) \prod_{m=1}^M \frac{f_m(a - (L+1))}{f_m(a + R)}$$

$$+ 2 p(\|z\|_\infty \leq R)^{-1} \sum_{L=a}^{2a-R-1} \tilde{p}(L) \prod_{m=1}^M \frac{f_m(L - a)}{f_m(a + R)}$$

$$+ \frac{p(\|z\|_\infty \geq 2a + R)}{p(\|z\|_\infty \leq R)} \prod_{m=1}^M \frac{f_m(a - R)}{f_m(a + R)}$$

$$\leq 2 p(\|z\|_\infty \leq R)^{-1} \sum_{L=R}^{a-1} \tilde{p}(L) \prod_{m=1}^M \frac{f_m(0)}{f_m(L + R + 1)}$$

$$+ 2 p(\|z\|_\infty \leq R)^{-1} \sum_{L=a}^{2a-R-1} \tilde{p}(L) \prod_{m=1}^M \frac{f_m(0)}{f_m(2a + R - L)}$$

$$+ o_{a\to\infty}(1)$$

$$\leq \frac{2}{p(\|z\|_\infty \leq R)} \left(\prod_{m=1}^M f_m(0) \right) \sum_{L=0}^{a-R-1} \frac{\tilde{p}(L + R) + \tilde{p}(2a - R - 1 - L)}{\prod_{m=1}^M f_m(L + 2R + 1)} + o_{a\to\infty}(1)$$

$$\leq \frac{2}{p(\|z\|_\infty \leq R)} \left(\prod_{m=1}^M f_m(0) \right) \sum_{L=0}^\infty \frac{p(L + R)}{\prod_{m=1}^M f_m(L + 2R + 1)} + o_{a\to\infty}(1).$$

By our assumption, this vanishes as $a \to \infty$ and $R \to \infty$ slowly enough. $\qquad\square$

E.4.1. ASSUMPTION E.6 IN THE EXPERIMENTS

In this section we verify that Assumption E.6 is satisfied in our experiments. The noise we choose is $f_m \sim \frac{1}{2} g(\text{Laplace})$ where $g(z_m) = z_m$ if $z_m \leq 1$ and $g(z_m) = \text{sgn}(z_m)|z_m|^{1.1}$ if $z_m \geq 1$. This has continuous density with $f_m(z_m) \propto (2z_m)^{-\frac{0.1}{1.1}} \exp\left(-|2z_m|^{\frac{1}{1.1}}\right)$ if $|z_m| \geq \frac{1}{2}$ and $f_m(z_m) \propto \exp\left(-|2z_m|\right)$ if $|z_m| \leq \frac{1}{2}$. and one can check that the derivative of the logarithm is increasing almost everywhere on $(0, \infty)$ and approaches 0. The derivative has a jump discontinuity at $\frac{1}{2}$ but one can easily check that for any $\frac{1}{2} > z_m > 0$,

$$(\log f)'(z_m) = -2 < -\frac{0.1}{1.1} - \frac{2^{1/1.1}}{1.1} = (\log f)'(1) \leq (\log f)'(z_m + 1).$$

In our simulation, variables X^n are made up of neural networks $\tilde{h}_n$ applied to Gaussian variables. Neural networks are Lipschitz, so each X^n is sub-Gaussian. This is also the case when conditioning since if $E[\exp(tX^{n2})] < \infty$ for some $t > 0$ then if we pick some set S, $E[E[\exp(tX^{n2})|(X^m)_{m\in S}]] = E[\exp(tX^{n2})] < \infty$ so $E[\exp(tX^{n2})|(X^m)_{m\in S}] < \infty$ for almost every value of $(X^m)_{m\in S}$. Therefore $p(L + R) \leq \sum_{n\notin S} p(|X^n| > L + R|(X^m)_{m\in S})$ decays with negative square exponential tails, much faster than $f_m(L + 2R - 1)$. For some $C > 0$, any set of positive numbers, $(h_m)_m$, and large enough R,

$$\sum_{L=0}^\infty \frac{p(L + R)}{\prod_{m=1}^M f_m(h_m(L + 2R + 1))} \lesssim \sum_{L=0}^\infty \frac{\exp(-C(L + R)^2)}{\exp(-(\sum_m h_m)(L + 2R))} = \sum_{L=0}^\infty \exp\left(-C(L + R)^2 + (L + 2R) \sum_m h_m\right).$$

This sum is clearly finite and for large enough R, each term is decreasing in R. Therefore, the sum converges to 0 as $R \to \infty$. Thus Assumption E.6 is satisfied in our experiments.

E.4.2. DISCUSSION OF THEOREM 4.3

Theorem 4.3 states that if we choose our noise densities $(f_m)_m$ to have thick tails then we can answer the separating set selection problem by answering the separating representation search problem and setting $\mathrm{SepSet}(X, Y) = \{Z_m\}_{\psi_m = 1}$. It however does not suggest that this is the only strategy for choosing $(f_m)_m$ and it does not say anything about when ψ_m can be between $(0, 1)$.

For the answers of the separating set selection problem and the separating representation problem to disagree we must have a $(\psi_m)_m$ such that for all measurable sets A, B and all values $(\tilde{z}_m)_m$

$$0 = p(X \in A, Y \in B | (\tilde{z}_m)_m, (\psi_m)_m, (f_m)_m) - p(X \in A | (\tilde{z}_m)_m, (\psi_m)_m, (f_m)_m) p(X \in A | (\tilde{z}_m)_m, (\psi_m)_m, (f_m)_m). \tag{8}$$

This is an infinite set of constraints that must be satisfied by a value of an M-dimensional parameter $(\psi_m)_m$.

In Example E.4 we showed that if the variables are jointly Gaussian and $(f_m)_m$ are also specially chosen to be Gaussian then there can be a separating representation but no separating set. In this case, two variables are conditionally independent if their conditional covariance, which does not depend on $(\tilde{z}_m)_m$, is 0. Thus the infinitely many constraints in Eq. 8 collapse to one:

$$\mathrm{Cov}[XY | \tilde{Z}_\psi] = 0.$$

We conjecture that for generic choices of $(f_m)_m$, the infinitely many constraints in Eq. 8 remain distinct; they are therefore impossible to satisfy for any value of ψ that is not close to the indicator of a separating set. In this case we conjecture $\{Z_m\}_{\psi_m^* > c}$ is a separating set for any value of $1 - \epsilon > c > \epsilon$.