Targeted Cause Discovery with Data-Driven Learning

Jang-Hyun Kim *janghyun@mllab.snu.ac.kr*
Department of Computer Science, Seoul National University

Claudia Skok Gibbs *csg337@nyu.edu*
Center for Data Science, New York University

Sangdoo Yun *sangdoo.yun@navercorp.com*
NAVER AI Lab

Hyun Oh Song *hyunoh@mllab.snu.ac.kr*
Department of Computer Science, Seoul National University

Kyunghyun Cho *kyunghyun.cho@nyu.edu*
Center for Data Science, New York University
Prescient Design, Genetech

Abstract

We propose a novel machine learning approach for inferring causal variables of a target variable from observations. Our goal is to identify both direct and indirect causes within a system, thereby efficiently regulating the target variable when the difficulty and cost of intervening on each causal variable vary. Our method employs a neural network trained to identify causality through supervised learning on simulated data. By implementing a local-inference strategy, we achieve linear complexity with respect to the number of variables, efficiently scaling up to thousands of variables. Empirical results demonstrate the effectiveness of our method in identifying causal relationships within large-scale gene regulatory networks, outperforming existing causal discovery methods that primarily focus on direct causality. We validate our model's generalization capability across novel graph structures and generating mechanisms, including gene regulatory networks of E. coli and the human K562 cell line. Implementation codes are available at `https://github.com/snu-mllab/Targeted-Cause-Discovery`.

1 Introduction

Identifying causality among variables is a fundamental problem in machine learning, with applications ranging from generative modeling to system explanation and variable control (Schölkopf, 2022). Conventional approaches for inferring causality from observations often rely on assumptions about generating processes and utilize independence tests or model fitting (Spirtes et al., 2001; Brouillard et al., 2020). However, these methods become impractical in large-scale systems with thousands of variables and complex generation mechanisms due to the exponential complexity of algorithms or invalidity of assumptions (Zanga et al., 2022).

In this study, we present a scalable method for identifying causes of target variables in large-scale complex systems, such as gene regulatory networks (GRNs) (Karlebach & Shamir, 2008). Our method aims to identify all causal variables of a target variable, both direct and indirect (Figure 1). For instance, in GRNs, identifying causal transcription factors of a target gene facilitates drug development, as it allows for the regulation of the target gene's expression (Huynh-Thu et al., 2010). On the other hand, regulating each causal transcription factor involves varying levels of difficulty and cost (Martin & Sung, 2018). By identifying not only direct causes but also all causal factors, our approach allows for the prioritization of transcription factors that are less costly and easier to intervene, thereby streamlining the development process. We refer to this problem setting as ***targeted cause discovery***.

It is worth noting that one can infer all causal variables given an estimated direct-causal graph of the system by traversing the ancestors. However, the imperfection of direct-causal discovery methods presents a problem: prediction errors exponentially propagate through the traversal. Due to this critical issue, existing direct-causal discovery methods fall short in our setting of identifying all causal variables.

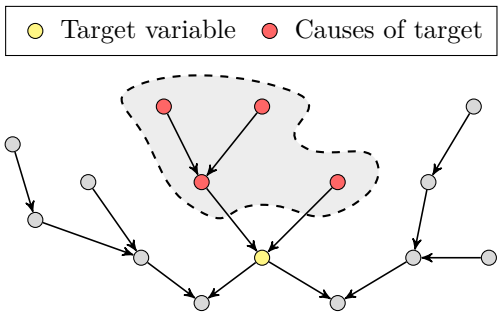


Figure 1: Illustration of targeted cause discovery in a causal graph.

Rather than inferring direct causality, we propose an end-to-end machine learning approach that identifies all causal variables of a target variable given observation data including interventions. Our method trains a deep neural network on simulated data to learn a causal discovery algorithm that generalizes to unseen causal structure (Ke et al., 2023). This data-driven learning approach eliminates the need for explicit assumptions and achieves reduced inference complexity compared to conventional causal discovery methods with exponential complexity (Zanga et al., 2022). We employ the Transformer architecture for our causal discovery model (Vaswani et al., 2017), building on its demonstrated effectiveness in previous works (Lorch et al., 2022). However, the quadratic complexity of the attention mechanism poses computational challenges in large-scale settings. To address this, we propose a novel local inference strategy, achieving linear inference complexity with respect to the number of variables and observations. We provide theoretical validation of our strategy in Section 4.

Empirical evaluations demonstrate our method's capability to identify causality in systems with thousands of variables, where existing causal discovery methods suffer from scalability issues even for systems of a hundred variables (Spirtes et al., 2001; Brouillard et al., 2020; Lorch et al., 2022). We apply our method to GRN simulation data, demonstrating that our model, trained on random causal-graph structures, effectively identifies causal relationships within GRNs of E. coli, yeast, and the K562 human cell line (Dibaeinia & Sinha, 2020; Replogle et al., 2022). We further explore the model's generalization capability using synthetic datasets, analyzing performance across different graph structures, generation mechanisms, and noise types, highlighting its potential for real-world applications.

2 Preliminary and Related Work

2.1 Causality

We consider a set of random variables $\mathcal{V} = \{\mathrm{x}_1, \ldots, \mathrm{x}_n\}$ that has a causal structure represented by a directed acyclic graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The joint distribution $p(\mathcal{V})$ is defined as $\prod_i p(\mathrm{x}_i \mid \mathrm{pa}(\mathrm{x}_i))$ where $\mathrm{pa}(\mathrm{x}_i)$ means the set of parent variables of x_i in $\mathcal{G}$. Through ancestral sampling, we obtain observations of variables.

Definition 1 (Causality)**.** *A variable* x_j *is a cause of* x_i *if and only if* $\exists c$ *s.t.*

$$p(\mathrm{x}_i \mid \mathrm{do}(\mathrm{x}_j = c)) \neq p(\mathrm{x}_i).$$

The operator $\mathrm{do}(\cdot)$ represents an intervention that fixes specific variables to predefined values during the ancestral sampling process (Pearl, 2009). Specifically, the observation for $\mathcal{V}$ under $\mathrm{do}(\mathrm{x}_j = c)$ follows $\delta_c(\mathrm{x}_j) \prod_{i \neq j} p(\mathrm{x}_i \mid \mathrm{pa}(\mathrm{x}_i))$, where δ_c is a Dirac delta function at c. Definition 1 implies that interventions on causal variables make a distributional change in the target variable.

We denote the set of causes of x_i as $\mathrm{ca}(\mathrm{x}_i)$. We refer to the problem of identifying all causal variables of a target variable as *targeted cause discovery*, distinguishing it from the conventional term causal discovery, which typically refers to the problem of determining the direct-causal graph structure.

2.2 Related Work

Conventional causal discovery approaches aim to infer the causal graph structure $\mathcal{G}$ from observations (Zanga et al., 2022). These approaches can be broadly categorized into constraint-based and score-based methods. Constraint-based approaches employ conditional independence testing and formalized directional decision

rules to identify causal relationships (Spirtes et al., 1995; 2001). These approaches require independence testing over combinatorial sets of variables, resulting in exponential complexity with respect to the number of variables (Colombo et al., 2014). Score-based approaches optimize the goodness-of-fit of graph structures on observations while balancing graph complexity (Chickering, 2002; Hauser & Bühlmann, 2012; Zheng et al., 2018; Brouillard et al., 2020). To navigate the combinatorial graph search space, these approaches rely on tailored assumptions about graph structures and generation mechanisms (Lopez et al., 2022). An alternative research direction involves estimating the topological ordering of causal variables to reduce optimization complexity (Reisach et al., 2021; Sanchez et al., 2023). Additionally, there are efforts to locally identify the direct causes of a specific variable through conditional independence testing, though these methods still face exponential complexity (Aliferis et al., 2010; Gao & Ji, 2015).

Recent efforts have shifted from traditional statistical modeling of graph structures and generation mechanisms to learning-based approaches that leverage large amounts of data and computational power (Lopez-Paz et al., 2015; Löwe et al., 2022; Lorch et al., 2022; Ke et al., 2023; Wu et al., 2024). These methods involve collecting synthetic data with known ground-truth causal graphs and training neural networks to predict graph structures, given the observations (Ke et al., 2023). In this work, we analyze the generalization performance of these data-driven approaches in large-scale, complex systems. Specifically, we propose a novel strategy that identifies both direct and indirect causes with linear complexity, efficiently scaling to accommodate thousands of variables.

3 Targeted Cause Discovery versus Causal Discovery

Targeted cause discovery offers several technical advantages over direct-causal discovery, suggesting that conventional causal discovery methods can be ineffective in scenarios where identifying all causes of a target is the primary objective.

1) Error propagation mitigation: Inferring causes from an inaccurately estimated direct-causal graph leads to exponentially propagated errors. For a direct-causal discovery algorithm with an expected prediction error rate of e, the error rate for estimating causes at a distance d from the target on the graph is approximately $1-(1-e)^d$. Our cause discovery method circumvents this issue by directly inferring causality among distant variables. Figure 2, shows empirical measurements of prediction error rates over varying distances between the target variable and its causes. The results demonstrate that our method maintains consistent error rates regardless of the distances, while error rates in the direct-causal discovery method increase with distance.

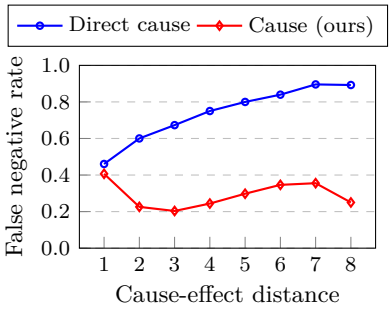


Figure 2: Targeted cause discovery error rate as a function of shortest path length between variables. *Direct cause* denotes an approach that infers causes from an estimated direct-causal graph, while *Cause* refers to our method that directly estimates causes of a variable. We provide detailed experimental setting in Appendix B.5.

2) Addressing sparsity: Targeted cause discovery mitigates the sparsity issue of identifying direct causes in large-scale settings. For example, in the E. coli GRN, there are approximately 2.3 direct causes per gene among 1,565 genes (Dibaeinia & Sinha, 2020). This severe sparsity leads to imbalanced classification and poses challenges for machine learning algorithms (Kaur et al., 2019). As shown in Table 1, converting the problem from identifying direct causes to identifying all causes alleviates this sparsity, thereby easing the associated technical challenges.

Table 1: Averaged ratio of causes (or direct causes) per node, *i.e.*, $|\text{ca(x)}|/n$ (or $|\text{pa(x)}|/n$). We draw statistics from 10 graphs, each with 1000 nodes and an average in-degree of 2.

Graph type	Direct cause	Cause
Erdős–Rényi	0.2%	1.1∼1.2%
Scale-free	0.2%	0.4∼2.2%
Gene regulatory	0.2%	0.5∼1.1%

3) Local inference guarantee: Targeted cause discovery theoretically ensures local inference, allowing for the inference of relationships between variables using only a subset of the system's variables (Proposition 1). This property is especially useful in large-scale settings where processing data from all variables becomes

computationally prohibitive. Leveraging this property, we propose an efficient algorithm capable of scaling to thousands of variables, as detailed in Section 4.2. It is worth noting that, as described in Proposition 1, the local inference does not hold for direct-causal discovery. We provide a proof in Appendix A.

Proposition 1. *For a variable subset* $V = \{\mathrm{x}_i\} \cup \{\mathrm{x}_j \mid j \in I\}$ *where* $I \subseteq \{1, \ldots, n\}$*, let* $\mathrm{ca}(\mathrm{x}_i; V)$ *indicate the set of causal variables of* x_i *in the system consisting of* V *(i.e., the causal graph with variables not in* V *marginalized out). Similarly, we define* $\mathrm{pa}(\mathrm{x}_i; V)$ *as the set of direct causes within* V*. Then,* $\mathrm{ca}(\mathrm{x}_i; V) = \mathrm{ca}(\mathrm{x}_i) \cap V$*. However, a counter-example exists for direct causes:* $\mathrm{pa}(\mathrm{x}_i; V) \neq \mathrm{pa}(\mathrm{x}_i) \cap V$*.*

4 Method

In this section, we present our method for Targeted Cause Discovery with Data-driven Learning, termed ***TCD-DL***. We consider a system with n variables $\{\mathrm{x}_1, \ldots, \mathrm{x}_n\}$ having an underlying causal structure $\mathcal{G}$. We denote the observation data as $X \in \mathbb{R}^{n \times m}$, where m is the number of observations. In the interventional setting, we define a boolean matrix $M \in \{0, 1\}^{n \times m}$, where 1 indicates the occurrence of interventions.

Our problem objective is as follows: Given a target variable x_i, predict a label $l_i \in \{0, 1\}^n$ from the observation X and intervention matrix M, where $l_i[j] = 1$ means x_j is a cause of x_i, *i.e.*, $\mathrm{x}_j \in \mathrm{ca}(\mathrm{x}_i)$. We approach this problem from a probabilistic view, estimating a continuous ***cause score*** vector $s_i \in \mathbb{R}^n$, where $s_i[j]$ measures the relative likelihood of x_j being a cause of x_i. Here, a higher score indicates a higher likelihood of causality. This continuous relaxation allows for the use of gradient-based optimization techniques, which are well-suited for large-scale settings (Bottou, 2010).

4.1 Data-Driven Learning

A straightforward approach to discovering causality involves conducting interventions on every single variable with a statistically sufficient number of trials to confirm the hypothesis. However, it is often impractical to intervene at every single variable due to experimental limitations and the high costs associated with conducting a sufficient number of trials (Addanki et al., 2020).

To address this challenge, we develop a parameterized model f_θ capable of inferring causality among variables from observations X of arbitrary size. Specifically, given the index i of the target variable, the model processes the entire dataset $X \in \mathbb{R}^{n \times m}$ with intervention matrix M and returns a cause score vector $s_i \in \mathbb{R}^n$ as

$$s_i = f_\theta(X, M, i). \tag{inference}$$

By leveraging the entire dataset, the model processes relational information among all variables.

We implement f_θ as a deep neural network and train this model on simulated data $\mathcal{D} = \{(X_k, M_k, \mathcal{G}_k) \mid k \in \mathcal{I}\}$ to learn a causal discovery algorithm that generalizes to unseen causal structure (Ke et al., 2023). Here, $\mathcal{I}$ denotes an index set and X_k represents an observation dataset sampled from a synthetic causal graph $\mathcal{G}_k$ with intervention matrix M_k. We generate synthetic datasets by using a simulator with predefined generation mechanisms on random graphs, as detailed in Section 5. For a causal graph $\mathcal{G}_k$ with n_k variables, we compute a label $l_{k,i} \in \{0, 1\}^{n_k}$ for each variable $i = 1, \ldots, n_k$, where $l_{k,i}[j] = 1$ means that the j-th variable is the cause of the i-th variable in $\mathcal{G}_k$. Our training objective is

$$\underset{\theta}{\text{minimize}} \;\; \mathbb{E}_{k \sim \mathcal{I}} \mathbb{E}_{i \sim \{1, \ldots, n_k\}} [\mathcal{L}(f_\theta(X_k, M_k, i), l_{k,i})], \tag{training}$$

where $\mathcal{L}$ is the loss function. In this study, we use binary cross-entropy with logits for $\mathcal{L}$ (Wei et al., 2022).

The model f_θ comprises two sequential modules, a feature extractor and a score calculator, designed to optimize compute efficiency. The feature extractor g_θ processes the stack $[X, M] \in \mathbb{R}^{n \times m \times 2}$ to produce features $F \in \mathbb{R}^{n \times 2 \times d}$ where each variable corresponds to two d-dimensional features. We employ a multi-layer axial Transformer for the feature extractor (Ho et al., 2019), a widely adopted architecture in prior studies (Lorch et al., 2022; Ke et al., 2023). In this study, we evaluate the basic form of the model architecture without positional encoding to ensure permutation equivariance, while noting that general model architectures for tensor inputs are applicable to our framework. We provide a detailed model architecture in Appendix B.1.

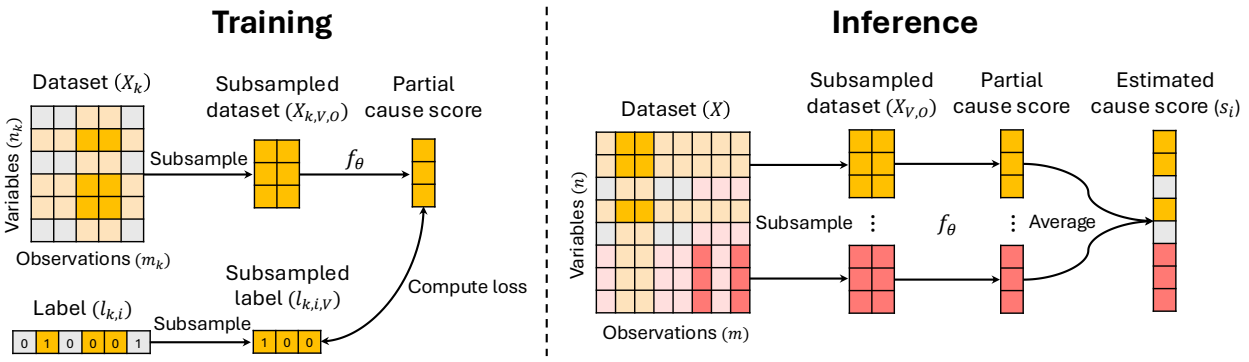


Figure 3: **Overview of our method.** The left figure illustrates a single training step. For simplicity, we exclude the intervention matrix in the figures, which is also an input to the model f_θ.

The score calculator h computes the cause score vector s_i for a target index i using dot-product between features as $h(F, i) = F[:, 0]F[i, 1] \in \mathbb{R}^n$, where $F[:, 0] \in \mathbb{R}^{n \times d}$ and $F[i, 1] \in \mathbb{R}^d$. To sum up, $f_\theta(X, M, i) = h(g_\theta(X, M), i) = h(F, i)$. This design allows for the reuse of feature F across multiple target indices i, enhancing training efficiency with batch data processing.

4.2 Local Inference for Scaling Up

The axial Transformer comprises three main operations: two attention layers over each variable and observation dimension, and a feed-forward layer (Ho et al., 2019). For an input $X \in \mathbb{R}^{n \times m}$, the complexities of attention layers are $O(n^2 m)$ and $O(nm^2)$, while the complexity of the feed-forward layer is $O(nm)$. The quadratic complexity of the attention mechanism poses challenges for large-scale data processing in terms of computational time and memory usage.

To address this issue, we propose a local inference strategy (Figure 3), supported by Proposition 1. Initially, we specify sizes n' ($< n$) and m' ($< m$) for variables and observations, respectively, suiting the computing resources. Given a target variable x_i, we randomly subsample a set of variables $V \subset \{\mathrm{x}_1, \ldots, \mathrm{x}_n\}$ with $|V| = n'$ and $\mathrm{x}_i \in V$. We then extract the corresponding observation matrix $X_V \in \mathbb{R}^{n' \times m}$ and intervention matrix $M_V \in \{0, 1\}^{n' \times m}$. Next, we subsample observations from X_V. However, some observations in X_V (*i.e.*, columns) may have unobserved intervened variables, which provide false causal signals among V. Thus, we select observations in X_V where no variables outside of V are intervened, and randomly subsample m' observations from the selected observations. We denote the resulting subsampled inputs as $X_{V,O} \in \mathbb{R}^{n' \times m'}$ and $M_{V,O} \in \{0, 1\}^{n' \times m'}$, where O denotes the set of subsampled observations. We refer to this subsampling process as $V, O \sim S(X, M, i)$.

We locally estimate the causality between variables in V and the target variable $\mathbf{x}_i$ as $f_\theta(X_{V,O}, M_{V,O}, i) \in \mathbb{R}^{n'}$. By aggregating and averaging the predictions over multiple subsamplings of variables and observations, we compute the entire cause score vector s_i. For $\mathrm{x}_j \in V$, let $f_\theta(X_{V,O}, M_{V,O}, i)[j]$ mean the cause score value of x_j to x_i, calculated with V and O. The ensembled estimation is then

$$s_i[j] = \mathbb{E}_{V,O \sim S(X,M,i)} \big[f_\theta(X_{V,O}, M_{V,O}, i)[j] \mid \mathrm{x}_j \in V \big]. \qquad \text{(ensembled local-inference)}$$

We provide details of the ensembling process in Algorithm 2. Our algorithm reduces the complexity from quadratic to linear with respect to the number of variables n. We leave the proof in Appendix A.

Proposition 2 (Algorithm complexity)**.** *Let n' and m' denote the subsampled variable and observation sizes, and let T denote the ensemble size. For a dataset X with n variables, the inference complexity of our algorithm is $O(nm'T(n' + m'))$.*

For training, we apply the identical random subsampling strategy on inputs and target labels. For X_k and a target variable index i, we denote the subsampled data as $X_{k,V,O} \in \mathbb{R}^{n', m'}$ and the corresponding target

label as $l_{k,i,V} \in \mathbb{R}^{n'}$. The local version of our training objective becomes

$$\underset{\theta}{\text{minimize}} \ \mathbb{E}_{k\sim\mathcal{I}}\mathbb{E}_{i\sim\{1,\dots,n_k\}}\mathbb{E}_{V,O\sim S(X_k,M_k,i)}[\mathcal{L}(f_\theta(X_{k,V,O}, M_{k,V,O}, i), l_{k,i,V})]. \qquad \text{(local training)}$$

We optimize f_θ using stochastic gradient descent with the AdamW optimizer (Loshchilov & Hutter, 2019). Algorithms 1 and 2 describe pseudo codes of our training and inference algorithms. We leave detailed hyperparameters in Appendix B.2.

Algorithm 1 Training (batch version)

inputs: $\mathcal{D} = \{(X_k, M_k, \mathcal{G}_k) \mid k \in \mathcal{I}\}$
parameters: subsample sizes n' and m', batch size b
initialize θ
repeat
  $\mathcal{X}, \mathcal{Y} \leftarrow \emptyset, \emptyset$
  for $j = 1$ to b **do**
    $k \leftarrow$ sample from $\mathcal{I}$
    $V, O \leftarrow$ subsample given X_k, M_k
    $\mathcal{X} \leftarrow \mathcal{X} \cup \{(X_{k,V,O}, M_{k,V,O}, i) \mid \mathrm{x}_i \in V\}$
    $\mathcal{Y} \leftarrow \mathcal{Y} \cup \{l_{k,i,V} \mid \mathrm{x}_i \in V\}$
  end for
  $\mathbf{g} \leftarrow$ calculate gradients $\nabla_\theta \mathcal{L}(f_\theta(\cdot), \cdot)$ on $\mathcal{X}, \mathcal{Y}$
  $\theta \leftarrow$ update using gradients $\mathbf{g}$
until convergence
return θ

Algorithm 2 Inference

inputs: $X \in \mathbb{R}^{n\times m}$, $M \in \{0,1\}^{n\times m}$, target index i
parameters: subsample sizes n' and m', #ensemble T
initialize $s_i \leftarrow \mathbf{0}_n$
for $t = 1$ to T **do**
  $I \leftarrow \text{permute}(\{1, \dots, n\} \setminus \{i\})$
  split $I = \cup_{j=1}^{b} I_j$ where $b = \lceil \frac{n}{n'} \rceil$ and $|I_j| \le n'$
  for $j = 1$ to b **do**
    $I_j \leftarrow I_j \cup \{i\}$ and $V \leftarrow \{\mathrm{x}_k \mid k \in I_j\}$
    $O \leftarrow$ subsample given V, X, M
    $s_i[I_j] \leftarrow s_i[I_j] + f_\theta(X_{V,O}, M_{V,O}, i)$
  end for
end for
$s_i \leftarrow s_i / T$
return s_i

5 Experiment

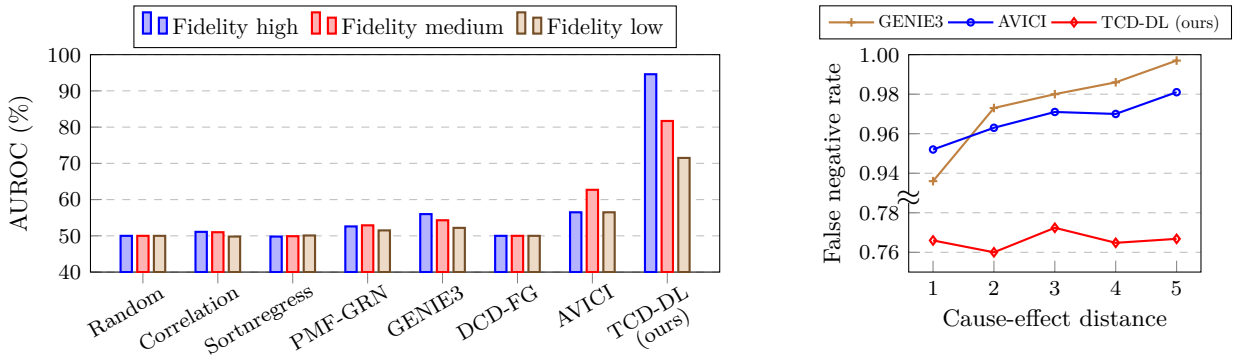
We provide experimental results for our method on targeted cause discovery. In Section 5.1, using a gene expression simulator, we verify that our model trained on random graphs effectively identifies causality in biological networks. In Section 5.2, we examine the generalization capability of our model across varying graph structures and generation mechanisms using synthetic datasets. In Section 5.3, we evaluate the impact of key design choices in our ensembled local-inference strategy.

5.1 Gene Regulatory Network

5.1.1 Setting

Simulated dataset. We generate the training set $\mathcal{D}$ and test set using the SERGIO GRN simulator (Dibaeinia & Sinha, 2020). This simulator produces single-cell gene expression data, modeling the stochastic nature of transcription based on a user-provided GRN. We conduct experiments over varying levels of simulator's observational fidelity, where higher fidelity yields expression data closer to the population parameters. Details about the simulator, including the generation mechanisms and the definition of fidelity levels, are provided in Appendix B.3.

For the training dataset, we use random graph structures including Erdős–Rényi (ER), Scale-Free (SF), and Stochastic Block Model (SBM) with 1,000 variables (Drobyshevskiy & Turdakov, 2019). The test data is generated from biological GRNs of E. coli (1,565 genes) and yeast (4,441 genes) as obtained from Marbach et al. (2009). Interventions are simulated by knocking out individual genes, *i.e.*, setting their transcription rates to zero. We generate 10 samples per intervention with the simulator configuration adopted from Lorch et al. (2022), along with 500 observational samples (details provided in Appendix B.3). We analyze the impact of intervention on performance in Appendix C.1.



(a) Performance across varying levels of simulator fidelity

(b) Prediction error analysis

Figure 4: **Benchmarking results.** (a) Performance on E. coli GRN with 1565 genes over varying levels of simulator's observational fidelity. We provide AUROC, AP, and F1 score values including standard deviations in Table 9. (b) Targeted cause discovery error rate as a function of shortest path length between variables.

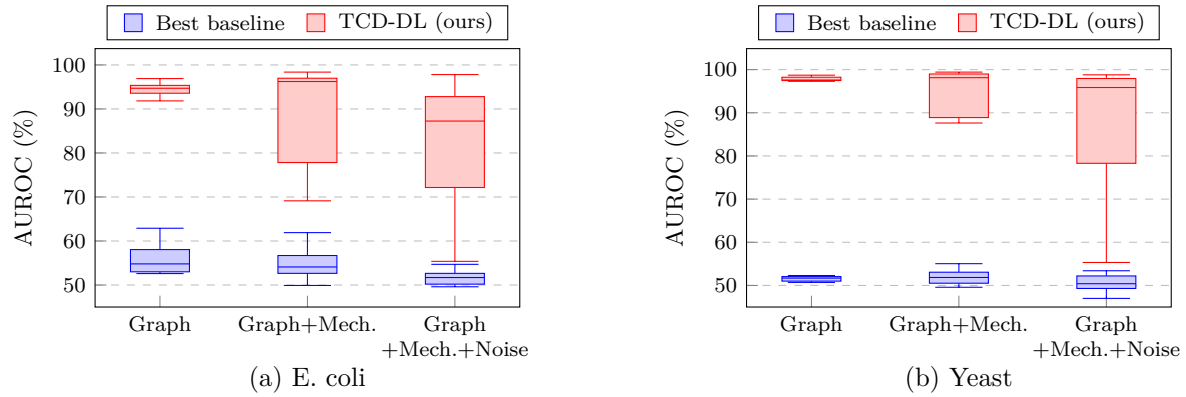
Baseline. We evaluate methods that are scalable to our setting, allowing for a comprehensive comparison across different approaches to causal discovery and GRN inference. As baselines, we include a *random* guessing model, the absolute *correlation* score, and the regression-based approach *sortnregress* (Reisach et al., 2021). For causal discovery methods discussed in Section 2.2, we consider the score-based approach *DCD-FG* (Lopez et al., 2022) and the learning-based approach *AVICI* (Lorch et al., 2022). Constraint-based methods like the PC algorithm are not considered in our analysis due to their limited scalability (Spirtes et al., 1995; 2001). Additionally, we assess GRN inference methods, including the tree-based method *GENIE3* (Huynh-Thu et al., 2010) and the linear factor-model *PMF-GRN* (Skok Gibbs et al., 2024). These methods compute likelihood scores for (direct) causal relationships between variable pairs. We derive cause score vectors s_i of baseline methods using these likelihood scores.

Evaluation metric. We evaluate the targeted cause discovery performance on variables having at least one causal variable. For each target variable, we compare the predicted cause score vector against the ground-truth binary label, where 1 indicates a causal relationship between variables. We employ AUROC, Average Precision (AP), and F1 score for this binary classification task (Rainio et al., 2024). To calculate the F1 score, we threshold the cause scores to match the number of positive predictions with the ground-truth labels. We measure average performance across all valid variables (those with at least one cause) in a test system. We obtain statistics using expression data with 10 different random seeds for sampling.

5.1.2 Analysis

Benchmarking results. Figure 4-a presents the targeted cause discovery performance on E. coli GRN. Our approach consistently achieves the best performance by a large margin, demonstrating the effectiveness of our data-driven learning. The results reveal the shortcomings of existing methods relying on specific assumptions. Linear models (correlation, sortnregress) fail to identify causality in our settings with complex generation mechanisms. As a sanity check, we observe that correlation achieves 70.7% AUROC on causal graphs with linear generation mechanisms, showing moderate performance under valid assumptions. The score-based approach (DCD-FG) performs near randomly, likely due to invalid assumptions about graph structures and generation mechanisms. Notably, our method significantly outperforms the existing learning approach (AVICI), demonstrating the effectiveness of our approach for targeted cause discovery.

To gain deeper insight into the performance improvements, we measure the false negative rate as a function of distance in the ground-truth causal graph (Figure 4-b). We threshold the cause scores of best-performing methods to ensure an identical number of positive predictions. The results reveal that cause scores of baseline methods exhibit an increasing error rate as the distance increases, while our method maintains consistent performance. These findings highlight a key advantage of our approach, identifying both proximal and distant causes without performance degradation.



(a) E. coli (b) Yeast

Figure 5: **Out-of-distribution performance.** Box plots of targeted cause discovery performance on novel graph structures, mechanism parameters (mech.), and noise configurations unseen during training.

Errors and identifiability. While our model demonstrates strong performance, there still remain errors, particularly as simulator fidelity decreases. To investigate the sources of these errors, we compare our model's performance on random graphs sampled from the training setting (*i.e.*, validation set) to its performance on E. coli graphs (*i.e.*, test set). Table 2 reveals that both validation and test performance decline as simulator fidelity decreases. This parallel degradation suggests that the primary source of error is not a generalization issue, but rather stems from other factors. This finding raises questions about causal identifiability in low-fidelity scenarios, indicating that the dataset itself may lack sufficient information for accurate causal inference.

Table 2: AUROC (%) on random graphs (validation) and E. coli GRN (test).

Data \ Fidelity	High	Medium	Low
Validation	89.6	83.3	70.1
Test	94.6	81.7	71.5

Human cell evaluation. We test our simulator-trained model in a real-world scenario using a Perturb-seq dataset derived from the K562 cell line of a patient with chronic myelogenous leukemia (Replogle et al., 2022). We focus on the gene MYC, a key oncogene involved in cell proliferation, growth, and apoptosis, frequently overexpressed in cancers (Dhanasekaran et al., 2022). We compute cause scores for 1,868 genes and select the top 20 genes as predicted causes of MYC expression (Appendix B.4). We compare these predicted causes against the STRING database (Szklarczyk et al., 2023), existing literature (Table 7), and top 20 genes with the highest expression correlations to MYC (Figure 6). Our model demonstrates strong predictive accuracy, achieving 90% precision compared to STRING and 30% precision against existing literature. Notably, only the gene EEF1A1 shows high expression correlation with the target, indicating that our method identifies novel causal factors not captured by correlation ranking. In Appendix C.3, we quantitatively compare our method to correlation ranking to support this claim, and provide additional analysis of our model on leukemia-related genes. These results highlight our model's potential for real-world applications in understanding and manipulating gene regulation, particularly in the context of personalized medicine and targeted therapies for cancer.

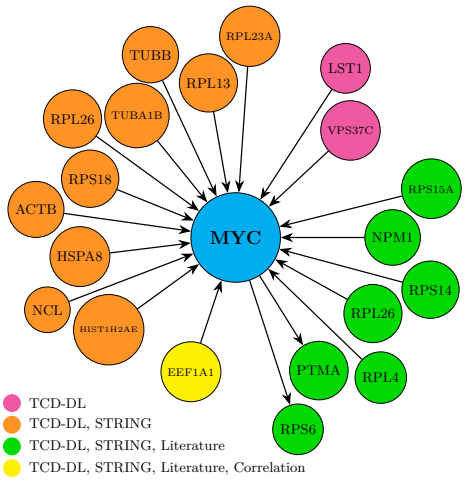
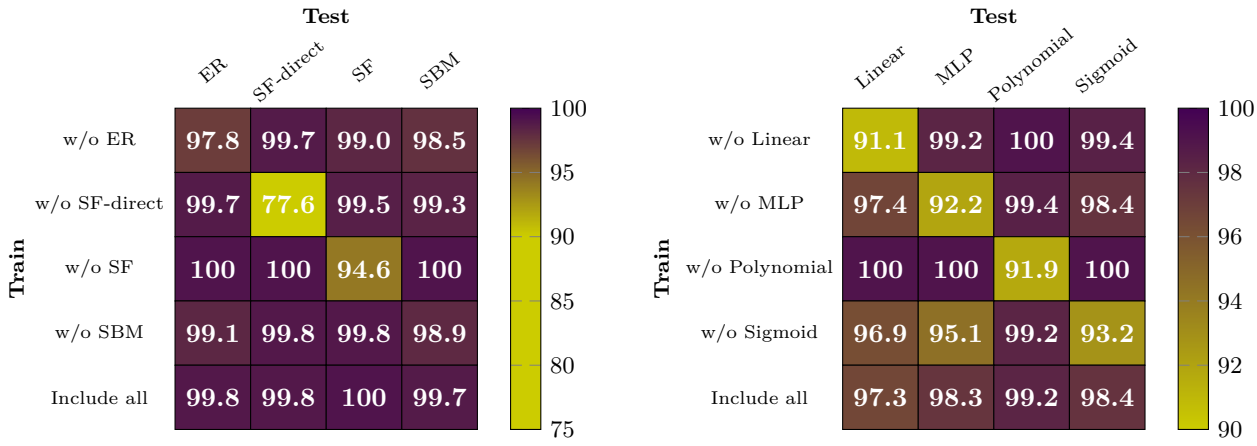


Figure 6: Causal genes of MYC identified by our method. Genes are categorized based on validation from STRING, existing literature, and top-20 gene expression correlations, where PTMA and RPS6 are validated as effects of MYC.

Out-of-distribution evaluation. We further study the generalization capabilities of our model by testing on mechanism and noise configurations that differ from the training. We adopt the configurations from Lorch



(a) Graph structure (b) Generation mechanism

Figure 7: **Training source ablation analysis.** The metric used is the relative AUROC, with 100 indicating the best model's performance on each test case and 0 corresponds to random prediction, *i.e.*, $100(p - p_{\text{random}})/(p_{\text{best}} - p_{\text{random}})$, where p denotes the AUROC score of a given model.

et al. (2022), as described in Appendix B.3. Figure 5 shows that our method largely outperforms the baselines across all settings, demonstrating robust generalization. Notably, when only the graph structure differs from training, our model shows high prediction capability. However, as generation mechanisms diverge from the training setting, the performance variance increases. These findings show both the strengths of our approach and the challenges inherent in generalizing to diverse generation mechanisms.

Runtime measurement. The runtime of our inference algorithm for processing each target variable is 2.5 seconds for E. coli (1,565 genes) and 7.8 seconds for yeast (4,441 genes), as measured with an NVIDIA RTX 3090 GPU. These results highlight that our model, with ensembled local inference, operates within seconds for large-scale systems. We provide a comparison of the runtime with baseline methods in Appendix C.2, where some baselines, such as DCD-FG and PMF-GRN, take several hours for inference.

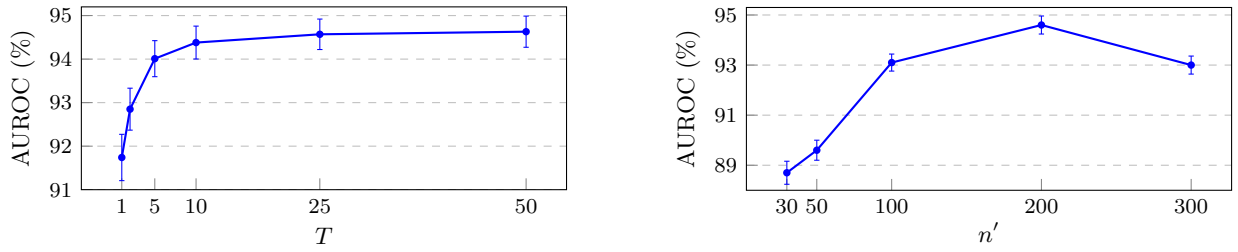
5.2 Ablation Study: Effect of Training Sources

To analyze the impact of training sources, we conduct ablation studies by removing specific causal graph types from the training dataset and evaluating the resulting models. In Figure 7-a, we use the simulator from Section 5.1 with varying graph structures, including Erdős–Rényi (ER), directional Scale-Free (SF-direct), Scale-Free (SF), and Stochastic Block Model (SBM) (Drobyshevskiy & Turdakov, 2019). For Figure 7-b, we modify the simulator's generation mechanism to include analytic functions (linear, non-linear multi-layer perceptron (MLP), polynomial, and sigmoid), while maintaining a scale-free network structure. We set function parameters according to Wu et al. (2024).

Figure 7 shows the relative performance ranging from 0 (random) to 100 (best) for each test case. Both subfigures exhibit similar patterns. Diagonal entries show lower performance, indicating a generalization gap between train and test data sources. However, relative performances exceed 90, demonstrating the strong generalization capability. Notably, models trained on all data types consistently achieve near-best performance. This data scaling effect mirrors observations in large-scale language models (Brown et al., 2020), indicating that diversifying training sources effectively enhances overall causal discovery performance.

5.3 Design Choice Analysis

We analyze the impact of design choices in our ensembled local-inference strategy by sweeping the ensemble size T and the subsampled variable size n' in Algorithm 2. Figure 8-a demonstrates that performance improves as ensemble size increases, plateauing around 25. This result validates the effectiveness of our



(a) Ensemble size (b) Subsampled input size

Figure 8: **Hyperparameter analysis.** Performance on E. coli GRN with varying (a) ensemble sizes T and (b) input variable subsample sizes n'.

ensembled inference approach. Figure 8-b illustrates the effect of subsampled variable size per input. Performance increases with input size up to 200, suggesting that processing larger variable sets through a single Transformer forward pass allows the model to utilize richer relational information. However, performance declines for input sizes exceeding 300, indicating conflicting effects between input complexity and information richness. These results validate our approach of processing a subset of variables, highlighting its effectiveness compared to processing all variables simultaneously.

6 Conclusion and Discussion

In this work, we propose an effective and scalable approach for targeted cause discovery, aiming to identify all causal variables of a target variable. Our approach trains a neural network that learns causal discovery algorithms from simulated data, offering an alternative to existing causal discovery approaches that rely on specific assumptions. To address large-scale systems, we introduce a local-inference strategy with theoretical guarantees. Our approach significantly outperforms causal discovery baselines on gene regulatory networks, demonstrating strong generalization capability across graph structures and generation mechanisms.

Our method shifts the focus from traditional explicit modeling of assumptions to a data engineering problem, aligning with recent successes of large-scale generative models. We anticipate further performance improvements through data scaling efforts with our scalable framework. On the other hand, the reliance on data and black-box models reduces interpretability. Ensuring causal identifiability while leveraging the strengths of data-driven methods represents an important future direction for targeted cause discovery.

Acknowledgement

The work was supported by the National Science Foundation (under NSF Award 1922658). Kyunghyun Cho is supported by the Samsung Advanced Institute of Technology (under the project Next Generation Deep Learning: From Pattern Recognition to AI). Jang-Hyun Kim and Hyun Oh Song are supported by SNU-NAVER Hyperscale AI Center and Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-00882, SW STAR LAB, Development of deployable learning intelligence via self-sustainable and trustworthy machine learning) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00354036). Claudia Skok Gibbs is supported by an NSF-GRFP award (DGE-2234660). Kyunghyun Cho is the corresponding author.

References

Raghavendra Addanki, Shiva Kasiviswanathan, Andrew McGregor, and Cameron Musco. Efficient intervention design for causal discovery with latents. *ICML*, 2020.

Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification. *Journal of Machine Learning Research*, 11, 2010.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. *COMPSTAT*, 2010.

Philippe Brouillard, Sébastien Lachapelle, Alexandre Lacoste, Simon Lacoste-Julien, and Alexandre Drouin. Differentiable causal discovery from interventional data. *NeurIPS*, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, et al. Language models are few-shot learners. *NeurIPS*, 2020.

Wenan Chen, Yan Li, John Easton, David Finkelstein, Gang Wu, and Xiang Chen. Umi-count modeling and differential expression analysis for single-cell rna sequencing. *Genome biology*, 19, 2018.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3, 2002.

Dominique Chu, Nicolae Radu Zabet, and Boris Mitavskiy. Models of transcription factor binding: sensitivity of activation functions to model assumptions. *Journal of Theoretical Biology*, 257, 2009.

Diego Colombo, Marloes H Maathuis, et al. Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 15, 2014.

Renumathy Dhanasekaran, Anja Deutzmann, Wadie D Mahauad-Fernandez, Aida S Hansen, Arvin M Gouw, and Dean W Felsher. The myc oncogene—the grand orchestrator of cancer growth and immune evasion. *Nature reviews Clinical oncology*, 19(1):23–36, 2022.

Payam Dibaeinia and Saurabh Sinha. Sergio: a single-cell expression simulator guided by gene regulatory networks. *Cell systems*, 11, 2020.

Mikhail Drobyshevskiy and Denis Turdakov. Random graph modeling: A survey of the concepts. *ACM computing surveys (CSUR)*, 52, 2019.

Ayako Egoh, Shin Nosuke Kanesashi, Chie Kanei-Ishii, Teruaki Nomura, and Shunsuke Ishii. Ribosomal protein l4 positively regulates activity of ac-myb proto-oncogene product. *Genes to Cells*, 15(8):829–841, 2010.

Brunangelo Falini, Lorenzo Brunetti, Paolo Sportoletti, and Maria Paola Martelli. Npm1-mutated acute myeloid leukemia: from bench to bedside. *Blood, The Journal of the American Society of Hematology*, 136 (15):1707–1721, 2020.

Tian Gao and Qiang Ji. Local causal discovery of direct causes and effects. *NeurIPS*, 2015.

Dalian Gong, Xinxu Rao, Ziqian Min, Xiaowen Liu, Huan Xin, Peijun Zhou, Lifang Yang, and Dan Li. Ube2s targets rpl26 for ubiquitination and degradation to promote non-small cell lung cancer progression via regulating c-myc. *American Journal of Cancer Research*, 13(8):3705, 2023.

Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research*, 13, 2012.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

Zhe Hong, Chengdang Xu, Shengfeng Zheng, Xinan Wang, Yiran Tao, Yao Tan, Guowen Lin, Denglong Wu, and Dingwei Ye. Nucleophosmin 1 cooperates with brd4 to facilitate c-myc transcription to promote prostate cancer progression. *Cell Death Discovery*, 9(1):392, 2023.

Vân Anh Huynh-Thu, Alexandre Irrthum, Louis Wehenkel, and Pierre Geurts. Inferring regulatory networks from expression data using tree-based methods. *PloS one*, 5, 2010.

Guy Karlebach and Ron Shamir. Modelling and analysis of gene regulatory networks. *Nature reviews Molecular cell biology*, 9, 2008.

Harsurinder Kaur, Husanbir Singh Pannu, and Avleen Kaur Malhi. A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM computing surveys (CSUR)*, 52, 2019.

Nan Rosemary Ke, Silvia Chiappa, Jane Wang, Anirudh Goyal, Jorg Bornschein, Melanie Rey, Theophane Weber, Matthew Botvinic, Michael Mozer, and Danilo Jimenez Rezende. Learning to induce causal structure. *ICLR*, 2023.

Mingli Li, Lu Yang, Anthony KN Chan, Sheela Pangeni Pokharel, Qiao Liu, Nicole Mattson, Xiaobao Xu, Wen-Han Chang, Kazuya Miyashita, Priyanka Singh, et al. Epigenetic control of translation checkpoint and tumor progression via ruvbl1-eef1a1 axis. *Advanced Science*, 10(17):2206584, 2023.

Jiayu Liang, Zhihong Liu, Zijun Zou, Xiangxiu Wang, Yongquan Tang, Chuan Zhou, Kan Wu, Fuxun Zhang, and Yiping Lu. Knockdown of ribosomal protein s15a inhibits human kidney cancer cell growth in vitro and in vivo. *Molecular Medicine Reports*, 19(2):1117–1127, 2019.

Yi-Te Lin, Hsing-Pang Lu, and Chuck C-K Chao. Oncogenic c-myc and prothymosin-alpha protect hepatocellular carcinoma cells against sorafenib-induced apoptosis. *Biochemical pharmacology*, 93(1):110–124, 2015.

Romain Lopez, Jan-Christian Hütter, Jonathan Pritchard, and Aviv Regev. Large-scale differentiable causal discovery of factor graphs. *NeurIPS*, 2022.

David Lopez-Paz, Krikamol Muandet, Bernhard Schölkopf, and Iliya Tolstikhin. Towards a learning theory of cause-effect inference. *ICML*, 2015.

Lars Lorch, Scott Sussex, Jonas Rothfuss, Andreas Krause, and Bernhard Schölkopf. Amortized inference for causal structure learning. *NeurIPS*, 2022.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *ICLR*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2019.

Sindy Löwe, David Madras, Richard Zemel, and Max Welling. Amortized causal discovery: Learning to infer causal graphs from time-series data. *Conference on Causal Learning and Reasoning*, 2022.

Daniel Marbach, Thomas Schaffter, Claudio Mattiussi, and Dario Floreano. Generating realistic in silico gene networks for performance assessment of reverse engineering methods. *Journal of computational biology*, 16, 2009.

Erik W Martin and Myong-Hee Sung. Challenges of decoding transcription factor dynamics in terms of gene regulation. *Cells*, 7, 2018.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Oona Rainio, Jarmo Teuho, and Riku Klén. Evaluation metrics and statistical tests for machine learning. *Scientific Reports*, 14, 2024.

Michael J Ravitz, Li Chen, Mary Lynch, and Emmett V Schmidt. c-myc repression of tsc2 contributes to control of translation initiation and myc-induced transformation. *Cancer research*, 67(23):11209–11217, 2007.

Alexander Reisach, Christof Seiler, and Sebastian Weichwald. Beware of the simulated dag! causal discovery benchmarks may be easy to game. *NeurIPS*, 34, 2021.

Joseph M Replogle, Reuben A Saunders, Angela N Pogson, Jeffrey A Hussmann, Alexander Lenail, Alina Guna, Lauren Mascibroda, Eric J Wagner, Karen Adelman, Gila Lithwick-Yanai, et al. Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq. *Cell*, 185(14):2559–2575, 2022.

Pedro Sanchez, Xiao Liu, Alison Q O'Neil, and Sotirios A Tsaftaris. Diffusion models for causal discovery via topological ordering. *ICLR*, 2023.

Bernhard Schölkopf. *Causality for machine learning*. Probabilistic and causal inference: The works of Judea Pearl, 2022.

Claudia Skok Gibbs, Omar Mahmood, Richard Bonneau, and Kyunghyun Cho. Pmf-grn: a variational inference approach to single-cell gene regulatory network inference using probabilistic matrix factorization. *Genome Biology*, 25, 2024.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2001.

Peter L Spirtes, Christopher Meek, and Thomas S Richardson. Causal inference in the presence of latent variables and selection bias. *UAI*, 1995.

Damian Szklarczyk, Rebecca Kirsch, Mikaela Koutrouli, Katerina Nastou, Farrokh Mehryary, Radja Hachilif, Annika L Gable, Tao Fang, Nadezhda T Doncheva, Sampo Pyysalo, et al. The string database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic acids research*, 51(D1):D638–D646, 2023.

Mathias Uhlén, Linn Fagerberg, Björn M Hallström, Cecilia Lindskog, Per Oksvold, Adil Mardinoglu, Åsa Sivertsson, Caroline Kampf, Evelina Sjöstedt, Anna Asplund, et al. Tissue-based map of the human proteome. *Science*, 347(6220):1260419, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. Mitigating neural network overconfidence with logit normalization. *ICML*, 2022.

Rachel B Wilson, Alexandra M Kozlov, Helia Hatam Tehrani, Jessica S Twumasi-Ankrah, Yun Jin Chen, Matthew J Borrelli, Cynthia G Sawyez, Siddhant Maini, Trevor G Shepherd, Robert C Cumming, et al. Elongation factor 1a1 regulates metabolic substrate preference in mammalian cells. *Journal of Biological Chemistry*, 300(3), 2024.

Menghua Wu, Yujia Bao, Regina Barzilay, and Tommi Jaakkola. Sample, estimate, aggregate: A recipe for causal discovery foundation models. *arXiv preprint arXiv:2402.01929*, 2024.

Paul E Young, Rashmi Kanagal-Shamanna, Shimin Hu, Guilin Tang, Beenu Thakral, Naval Daver, Ghayas C Issa, L Jeffrey Medeiros, and Sergej Konoplev. Chronic myeloid leukemia, bcr-abl1-positive, carrying npm1 mutation–first case series from a single institution. *Leukemia Research*, 111:106685, 2021.

Alessio Zanga, Elif Ozkirimli, and Fabio Stella. A survey on causal discovery: theory and practice. *International Journal of Approximate Reasoning*, 151, 2022.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *NeurIPS*, 2018.

Xiang Zhou, Qian Hao, Jun-ming Liao, Peng Liao, and Hua Lu. Ribosomal protein s14 negatively regulates c-myc activity. *Journal of Biological Chemistry*, 288(30):21793–21801, 2013.

A Proofs and Definitions

Proposition 1. *For a variable subset $V = \{\mathrm{x}_i\} \cup \{\mathrm{x}_j \mid j \in I\}$ where $I \subseteq \{1, \ldots, n\}$, let $\mathrm{ca}(\mathrm{x}_i; V)$ indicate the set of causal variables of x_i in the system consisting of V (i.e., the causal graph with variables not in V marginalized out). Similarly, we define $\mathrm{pa}(\mathrm{x}_i; V)$ as the set of direct causes within V. Then, $\mathrm{ca}(\mathrm{x}_i; V) = \mathrm{ca}(\mathrm{x}_i) \cap V$. However, a counter-example exists for direct causes: $\mathrm{pa}(\mathrm{x}_i; V) \neq \mathrm{pa}(\mathrm{x}_i) \cap V$.*

Proof. Definition 1 does not rely on what other variables are included in the current system. Thus by definition, we have $\mathrm{ca}(\mathbf{x}_i; V) = \mathrm{ca}(\mathbf{x}_i) \cap V$. In the case of direct cause, let us consider a chain with three variables of $\mathbf{x}_1 \to \mathbf{x}_2 \to \mathbf{x}_3$. By definition, $\mathrm{pa}(\mathbf{x}_3) = \{\mathbf{x}_2\}$. In the system of $V = \{\mathbf{x}_1, \mathbf{x}_3\}$, $\mathbf{x}_1$ becomes the direct cause of $\mathbf{x}_3$, *i.e.*, $\mathrm{pa}(\mathbf{x}_3; V) = \{\mathbf{x}_1\}$. On the other hand, $\mathrm{pa}(\mathbf{x}_3) \cap V = \emptyset$. Thus, $\mathrm{pa}(\mathbf{x}_i; V) \neq \mathrm{pa}(\mathbf{x}_i) \cap V$. □

Proposition 2 (Algorithm complexity)**.** *Let n' and m' denote the subsampled variable and observation sizes, and let T denote the ensemble size. For a dataset X with n variables and m observations, the inference complexity of our algorithm is $O(nm'T(n' + m'))$.*

Proof. The set of n variables can be partitioned into $\lceil \frac{n}{n'} \rceil$ inputs. For each input, the complexities of attention layers are $O(n'^2 m')$ and $O(n' m'^2)$, while the complexity of the feed-forward layers is $O(n'm')$. Thus processing each variable once results in a complexity of $O(\frac{n}{n'}(n'^2 m' + n' m'^2)) = O(nm'(n' + m'))$. Considering an ensemble size of T, the overall computational complexity becomes $O(nm'T(n' + m'))$. □

B Experimental Settings

B.1 Model Architecture

For feature extractor g_θ, we utilize an axial Transformer without positional encoding to ensure permutation equivariance (Ho et al., 2019). The architecture code is based on the implementation provided by Wu et al. (2024). Each Transformer layer comprises two attention layers, one along the variable dimension and one along the observation dimension, followed by a feed-forward layer. Both attention and feed-forward layers include layer normalization and a skip connection (Ba et al., 2016). Detailed configuration is provided in Table 3.

As described in Section 4.1, the input to the Transformer is a stack $[X, M] \in \mathbb{R}^{n \times m \times 2}$, and the output is a feature matrix $F \in \mathbb{R}^{n \times 2 \times d}$. In a basic axial Transformer, the output size for an input of size $n \times m \times 2$ is $n \times m \times d$, where d is the embedding dimension. We denote this output as H. To obtain the feature matrix of size $n \times 2 \times d$, we first average the output H of size $n \times m \times d$ along the observation dimension m, resulting in a matrix of size $n \times d$. We then apply two feed-forward layers and concatenate the results to produce the feature matrix F of size $n \times 2 \times d$. The total number of trainable parameters of our model is 62k. We will release the code for our model as open-source.

Table 3: Architecture configuration.

Argument	Value
Number of Transformer layers	10
Embedding dimension	16
Number of attention heads	16
Feed-forward layer hidden dimension	96

B.2 Hyperparameter and Computing Environment

Training hyperparameter. We train a neural network using the AdamW optimizer (Loshchilov & Hutter, 2019). The training hyperparameters are detailed in Table 4. We set the batch size to fit our GPU memory (24GB). We run training for 40,000 steps, with early stopping if validation accuracy does not improve over

4,000 steps, measured every 200 steps. We tune the learning rate among [6e-4, 8e-4, 1e-3], finding that 8e-4 provides the most stable training performance across all experimental settings. We observe that the optimal learning rate depends on the model configuration rather than the data types. We employ a cosine learning rate scheduler, which reduces sensitivity to learning rates during training (Loshchilov & Hutter, 2017). We do not use dropout, as it empirically reduces performance in our setting. We hypothesize that local training and inference, predicting answers with partial information, mitigate the need for dropout.

Table 4: Training hyperparameter.

Argument	Value
Batch size	32
Training step	40,000
Learning rate	8e-4
Learning rate scheduler	cosine
Weight decay	1e-5

Inference hyperparameter. Figure 8 describes the hyperparameters required for our inference procedure (Algorithm 2). Hyperparameters are selected based on the analysis in Section 5.3. Note that we use the same sizes, n' and m', during training. The observation size is set to 200, identical to the variable size, to fit within our GPU memory constraints during training.

Table 5: Inference hyperparameter.

Argument	Value
Subsampled variable size n'	200
Subsampled observation size m'	200
Ensemble size T	50

Computing environment. We conduct all experiments including training and inference, using a NVIDIA RTX 3090 GPU with 24GB memory. The training time for models in Section 5.1 is approximately 9h, while inference takes about a few seconds per target (Table 8).

B.3 GRN Simulator

In this section, we describe the SERGIO GRN simulator used in our experiments (Dibaeinia & Sinha, 2020). Given a user-defined GRN, the simulator generates a gene expression matrix based on a specified cell type configuration.

Generation mechanism. The simulator samples gene expressions from the steady state of a dynamic system modeled as stochastic differential equations (Dibaeinia & Sinha, 2020). Master regulators (*i.e.*, root nodes in the causal graph) operate independently without external regulatory inputs, evolving with fixed production and decay rates. The regulatory influence of each gene is represented by a Hill function with pre-determined interaction parameters (Chu et al., 2009). This mechanism captures non-linear relationships and time-lagged effects, providing a realistic model of gene behavior.

Technical noise. The simulator produces datasets that reflect the statistical properties of real-world single-cell experimental data, incorporating several types of measurement errors and technical noise:

1. Dropouts: A high proportion of gene expressions (typically 60-95%) are randomly set to zero, simulating the dropout effect common in single-cell technologies.
2. Outlier genes: Some genes are assigned unusually high expression levels, replicating the presence of outliers.

3. Library size: The total expression level for each cell (known as library size) follows a log-normal distribution, reflecting the variability.

The simulator applies these technical noises sequentially to the expression data sampled from the stochastic differential equations.

Observational fidelity. To generate the unique molecular identifier (UMI) count expression matrix, a quantification scheme in single-cell RNA-sequencing, the simulator applies Poisson random sampling to the expression values λ after incorporating the technical noises (Chen et al., 2018). That is, the final observation value v is derived as $v \sim \text{Poisson}(\lambda)$. To evaluate the impact of this Poisson sampling process on targeted cause discovery performance and to determine the performance ceiling of our method, we control the fidelity of the Poisson sampling and define three levels:

1. High fidelity: Uses expression λ directly as the observation.
2. Medium fidelity: Uses the mean of 100 samples drawn from $\text{Poisson}(\lambda)$.
3. Low fidelity: Uses a single sample drawn from $\text{Poisson}(\lambda)$.

In Figure 4, we analyze the performance differences across varying levels of the simulator's observational fidelity. Unless otherwise specified, we use the high-fidelity setting for our analysis.

Intervention. We use the interventional setting identical to Lorch et al. (2022). Specifically, we perform gene knockout by setting the expression level of a specific gene to zero. We sample 10 intervened observations per gene. For example, given a GRN with 1000 genes, this results in an interventional dataset with 10,000 observations. From these observations and variables, we randomly sample subsets of size $n' = 200$ and $m' = 200$, as provided in Table 5.

Dataset. This paragraph summarizes the configurations of datasets used in our experiments in Table 6, adopted from Lorch et al. (2022). We generate training data using random graphs while testing on biological GRNs, E. coli (1,565 genes) and yeast (4,441 genes), obtained from Marbach et al. (2009). Figure 9 shows the degree histograms of these graph structures, highlighting the different patterns between biological graphs and random graphs. We generate interventional data by performing gene knockout on each gene, obtaining 10 observations per intervention. For yeast, we obtain 5 observations per intervention due to its larger gene count. We include 500 observational data points, conducting inference using a mixture of observational and interventional data. We preprocess each observation matrix using $\log_2$ counts-per-million (CPM) normalization following previous works (Lorch et al., 2022).

For the out-of-distribution (OOD) analysis in Figure 5, we adopt configurations from Lorch et al. (2022) (see Table 4). These configurations include different mechanism function parameters, such as Hill function coefficients and decay rates, which differ from the training settings. We also test OOD technical noise types, as described in Table 6. These noise types reflect the statistics of different experimental datasets, which have varying dropout percentages, outlier ratios, and library size distributions (Lorch et al., 2022).

B.4 Human Cell Dataset

This section describes a Perturb-seq dataset used in the human cell experiments (Figure 6). The dataset contains gene expression data from the K562 cell line, which is derived from a patient with chronic myelogenous leukemia (Replogle et al., 2022). The dataset includes both interventional and observational data on gene expression. The intervention is performed through gene knockouts, identical to our simulation setting. From the Perturb-seq dataset, we obtain 1,868 genes that have undergone intervention. We conduct the inference among these genes. For each intervention, we randomly subsample 10 observations. We also sample 500 observational data points, consistent with the number used in our training setting (Table 6). Using this subsampled dataset, we run our TCD-DL inference algorithm to obtain cause scores for the target gene MYC, applying the same CPM normalization scheme used in our simulation data (Appendix B.3).

Table 6: **Dataset configuration.** Note for abbreviations used: ER (Erdős–Rényi), SF (Scale-Free), SF-direct (directional Scale-Free), and SBM (Stochastic Block Model) (Drobyshevskiy & Turdakov, 2019). For the training data, we randomly select the graph structure and edge degree independently from the candidate sets. We use a slash (/) symbol to separately denote the statistics for E. coli and yeast GRNs.

Argument	Training set	Test set
Graph structure	{ER, SF, SF-direct, SBM}	E. coli/yeast
Average edge degree	{2,4,6}	2.3/2.1
Dataset size $\lvert\mathcal{D}\rvert$ per graph structure	150	10
Variable size (n)	1,000	1,565/4,441
Number of observations per intervention	10	10/5
Observation size (interventional)	10,000	15,650/22,205
Observation size (observational)	500	500
Number of cell types	10	10
Technical noise type	10x-chromium	10x-chromium
Technical noise type (OOD)	-	{illumina, drop-seq, smart-seq}

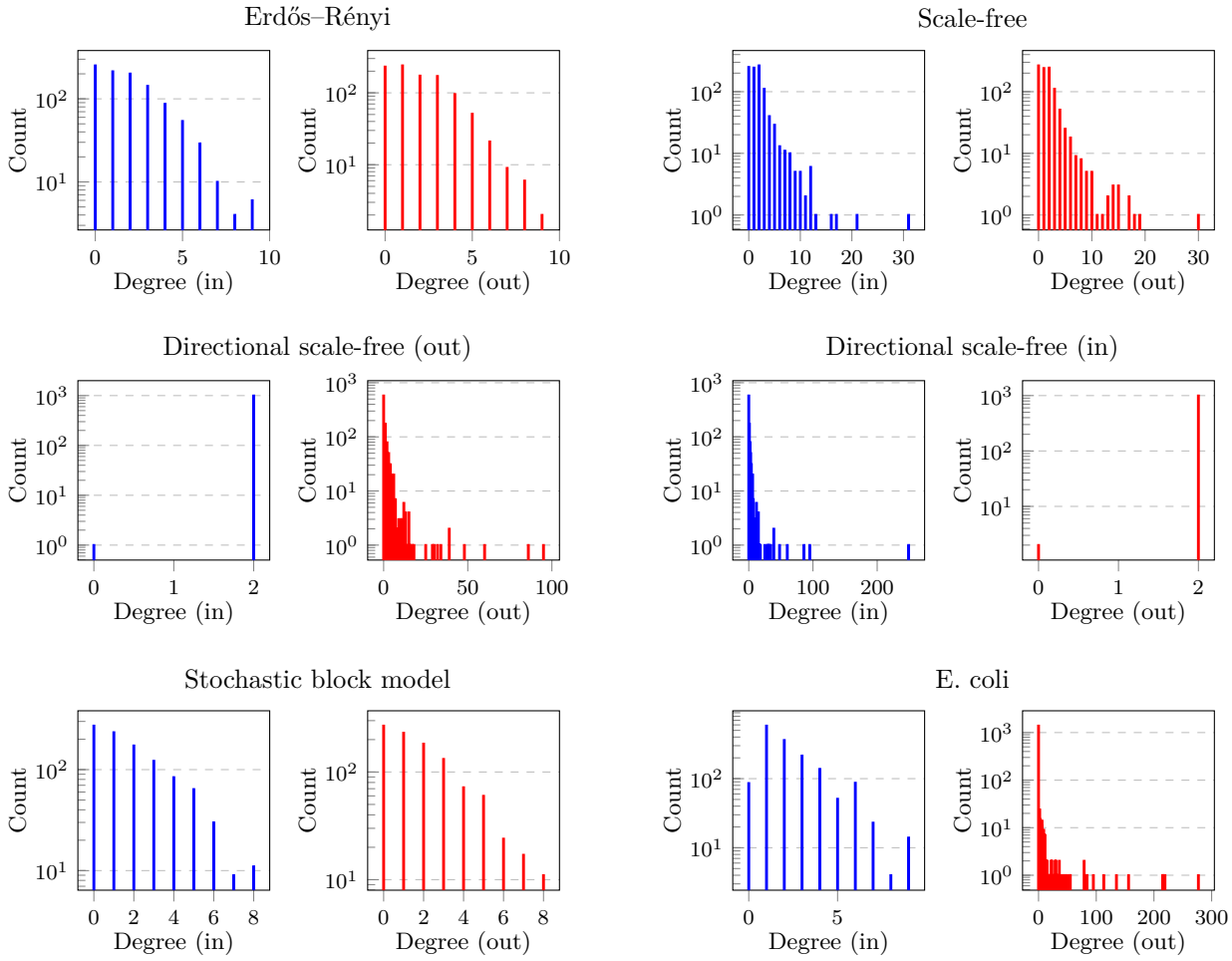


Figure 9: **Edge degree histograms** of graph structures considered in our experiments. All graphs have an average degree of 2. Blue represents the in-degree histogram, and red represents the out-degree histogram.

Table 7: Causal gene interactions for selected target gene MYC.

Target gene	Causal gene	Supported by
MYC	EEF1A1	TCD-DL, STRING, Literature (Li et al., 2023), (Wilson et al., 2024), Correlation
	NPM1	TCD-DL, STRING, Literature (Hong et al., 2023)
	PTMA	TCD-DL, STRING, Literature (Lin et al., 2015)
	RPL26	TCD-DL, STRING, Literature (Gong et al., 2023)
	RPL4	TCD-DL, STRING, Literature (Egoh et al., 2010)
	RPS14	TCD-DL, STRING, Literature (Zhou et al., 2013)
	RPS15A	TCD-DL, STRING, Literature (Liang et al., 2019)
	RPS6	TCD-DL, STRING, Literature (Ravitz et al., 2007)
	RPL23A	TCD-DL, STRING
	NCL	TCD-DL, STRING
	HSPA8	TCD-DL, STRING
	HIST1H2AE	TCD-DL, STRING
	RPL13	TCD-DL, STRING
	ACTB	TCD-DL, STRING
	TUBB	TCD-DL, STRING
	TUBA1B	TCD-DL, STRING
	RPL26	TCD-DL, STRING
	RPS18	TCD-DL, STRING
	LST	TCD-DL
	VPS37C	TCD-DL

B.5 Setting for Error Propagation Analysis

This section outlines the experimental setup for Figure 2 (Section 3). We generate scale-free random graphs comprising 100 nodes and set the causal mechanism using a 2-layer perceptron with a Tanh activation function (Wu et al., 2024). The root variables follow a Uniform distribution. We estimate the direct causal graph using the method proposed by Wu et al. (2024). To facilitate a clear comparison of how the false negative rate (FNR) changes with increasing cause-effect distance, we threshold the cause scores to achieve similar FNRs at a cause-effect distance of 1.

C Additional Experimental Results

C.1 Impact of Intervention

We quantitatively assess how intervention information impacts model performance. To this end, we conduct inference by replacing some of the interventional samples in the model's input $X_{V,O}$ with observational samples (Algorithm 2). Note, our model achieves an AUROC/AP of 94.5/38.6% on the E. coli GRN. In the first experiment, we replace samples with interventions performed on the ground-truth causes of the target gene with the observational samples. The inference results show a performance of 94.4/37.5%. Interestingly, we observe a slight decrease in performance, indicating that despite the absence of intervention on the ground-truth causes, our method leverages intervention information from other variables to amortize inference. On the other hand, when the input $X_{V,O}$ consists solely of observational samples, the performance dropped to 63.2/1.2%, demonstrating the limitations of inferring causality using only observational data.

C.2 Runtime Comparison

Table 8 compares the inference times of methods for targeted cause discovery. Some baseline methods (sortnregress and GENIE3) can individually compute a cause score vector for a target, while other baselines (AVICI, DCD-FG, PMF-GRN) require the calculation of the entire $n \times n$ score matrix to obtain a cause score vector for a target. From the table, our method conducts inference within seconds even for the yeast gene regulatory network (GRN) comprising 4,441 genes. In contrast, certain baselines encounter memory issues (AVICI) or require substantial computation time (DCD-FG, PMF-GRN).

Table 8: **Runtime measurement.** We measure inference time for identifying the causes of a target variable with an NVIDIA RTX 3090 GPU. *OOM* refers to the GPU out-of-memory error.

Species	Sortnregress	GENIE3	AVICI	DCD-FG	PMF-GRN	TCD-DL (ours)
E. coli (1,565 genes)	0.9s	1.7s	3.1s	1h 55m	3h 2m	2.5s
yeast (4,441 genes)	2.8s	2.7s	OOM	98h 31m	10h 46m	7.8s

C.3 Analysis on Human Cells

Comparison to correlation ranking. As illustrated in Figure 6, our method identifies novel causal factors for the gene MYC that are not captured by correlation ranking. To quantify the disparity between our model's predictions and correlation ranking, we analyze statistics across 1,868 genes from the Perturb-seq dataset. The average rank correlation between our model's cause scores and correlation-based rankings is 0.068, indicating low ranking similarity. When comparing the top 20 predictions from each approach, on average only 1.1 genes appear in both sets. Notably, an average of 6.4 genes from our top 20 predictions are validated by the STRING database, underscoring that our model identifies novel causal factors not captured by correlation.

Identifying causes of leukemia-related genes. To further validate our predictions from the K562 Perturb-seq dataset, we use the Human Protein Atlas to identify a set of 29 target genes associated with leukemia (Uhlén et al., 2015). We identify the top 10 causal predictions for each of these target genes, including those with support from the STRING database, and visualize these interactions in Figure 10. We further present the resulting GRNs for each leukemia target gene and its predicted causal regulators through network diagrams in Figures 11 and 12. These visualizations highlight the potential regulatory roles of our identified causal genes, providing insights into the predicted interactions driving leukemia. Validation against the STRING database demonstrates that our approach generates well-supported and highly relevant causal predictions.

Predicted influence of causal genes. We investigate the regulatory influence of each predicted causal gene over leukemia-related target genes in Figure 13. Notably, nucleophosmin (NPM1) is predicted to regulate 25 of the 29 leukemia target genes, receiving support from the STRING database for 21 of these predictions. NPM1 mutations are prevalent in approximately one-third of adult Acute Myeloid Leukemia (AML) cases, leading to an abnormal cytoplasmic localization of the NPM1 protein (Falini et al., 2020). Although NPM1 mutations are primarily associated with AML, recent studies have identified them in a small subset of Chronic Myeloid Leukemia (CML) patients (Young et al., 2021). The prediction that NPM1 regulates a substantial number of leukemia-associated target genes is particularly significant as it provides insights into potential key regulatory mechanisms underlying leukemia pathology. Understanding how NPM1 influences these target genes in CML could reveal critical pathways involved in leukemia progression and help identify novel therapeutic targets.

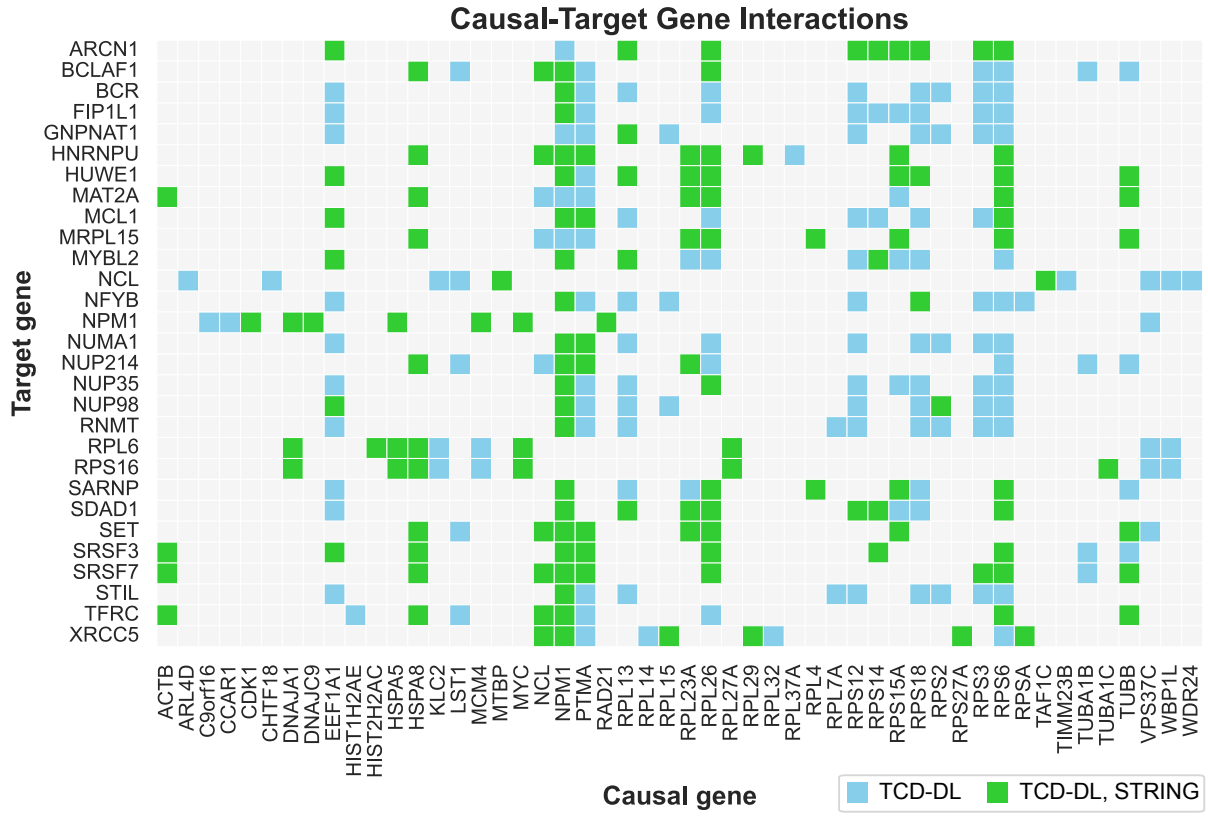


Figure 10: **Predicted causes of leukemia.** The matrix illustrates the predicted causality of leukemia-related target genes by TCD-DL predictions (blue) and TCD-DL predictions supported by STRING (green).

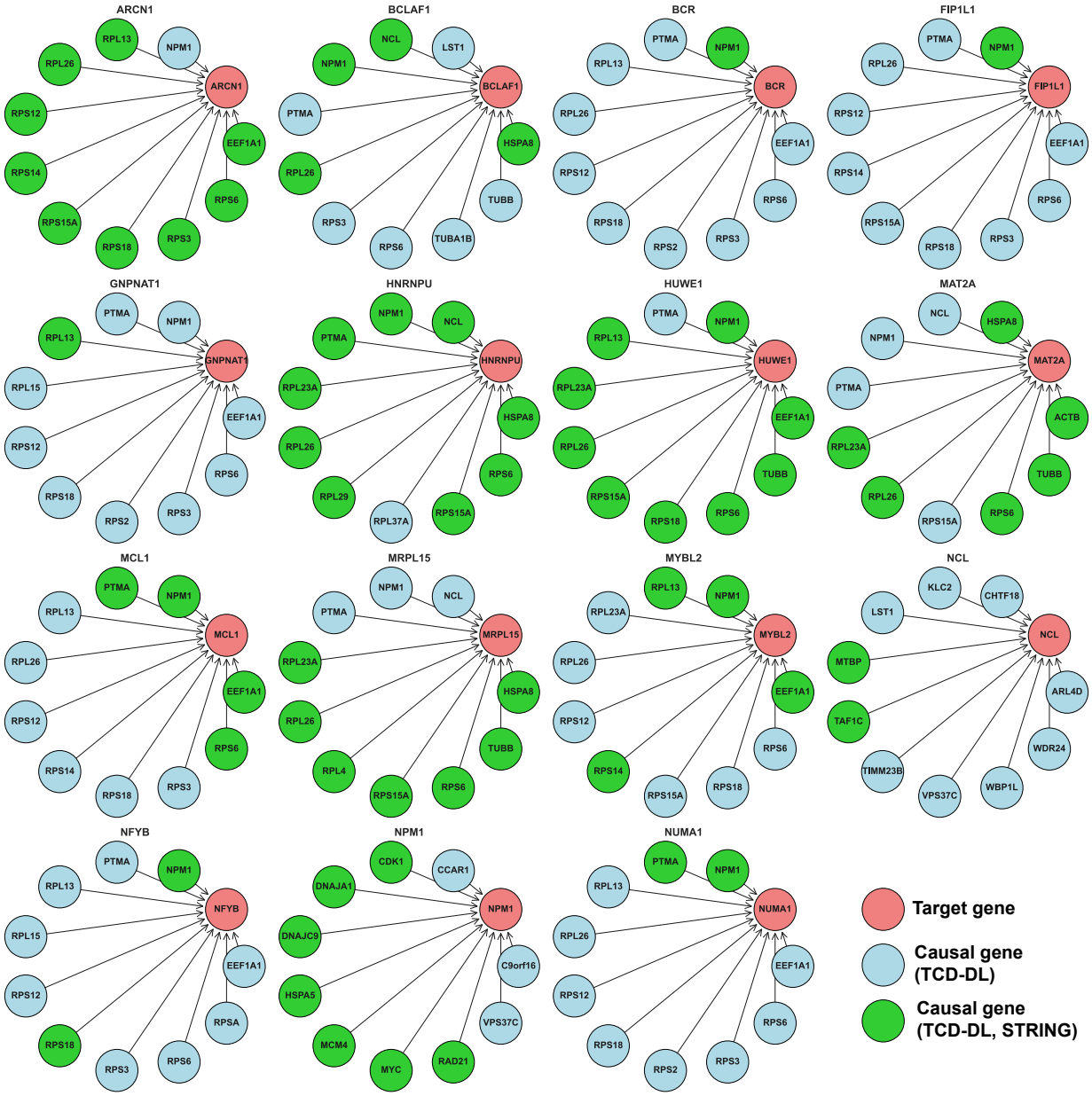


Figure 11: **Predicted causes of a leukemia-related gene.** GRNs illustrate the causal genes predicted by TCD-DL (blue) and TCD-DL predictions supported by STRING (green) for each leukemia-related target gene (pink).

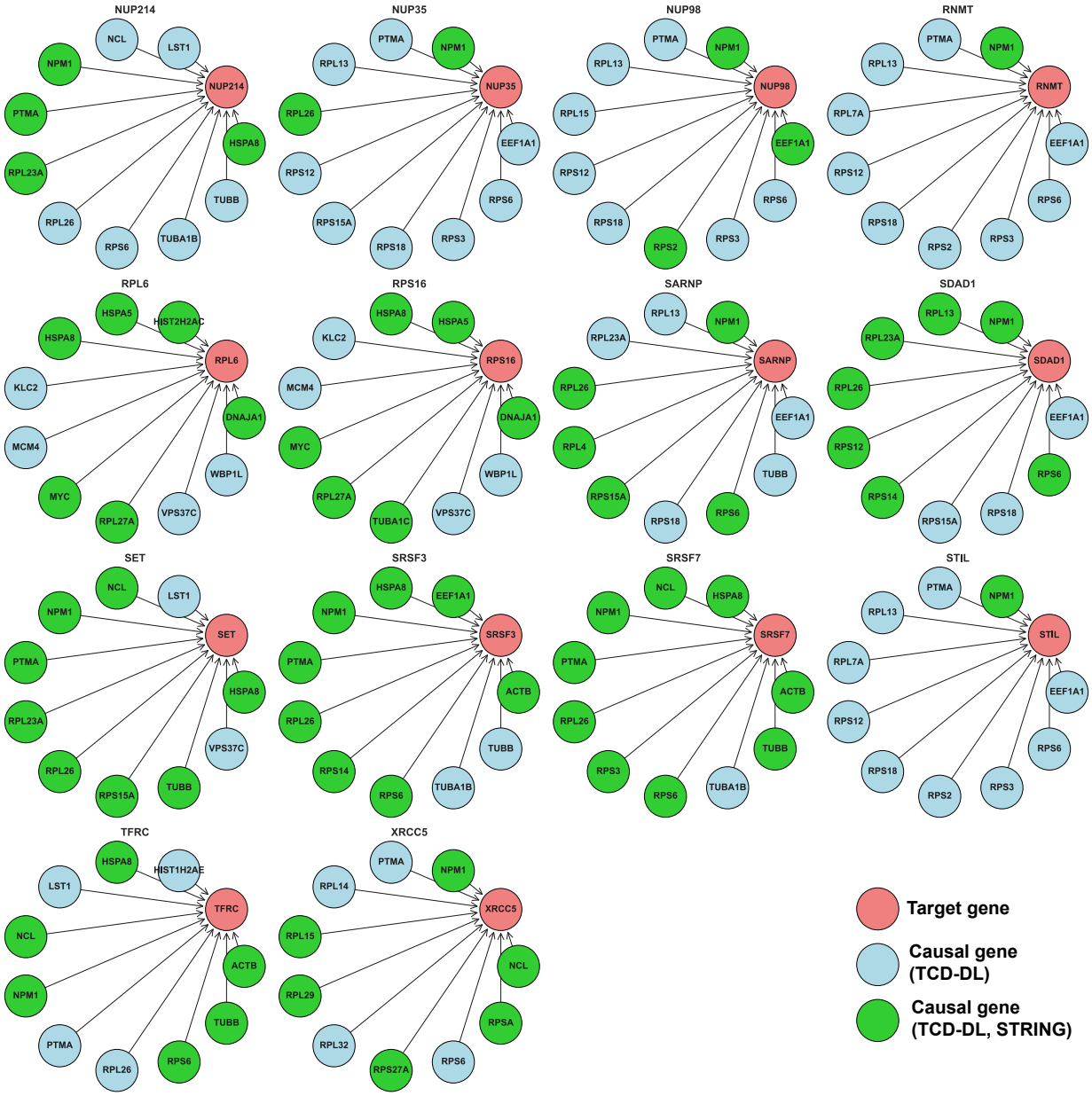


Figure 12: **Predicted causes of a leukemia-related gene.** GRNs illustrate the causal genes predicted by TCD-DL (blue) and TCD-DL predictions supported by STRING (green) for each leukemia-related target gene (pink).

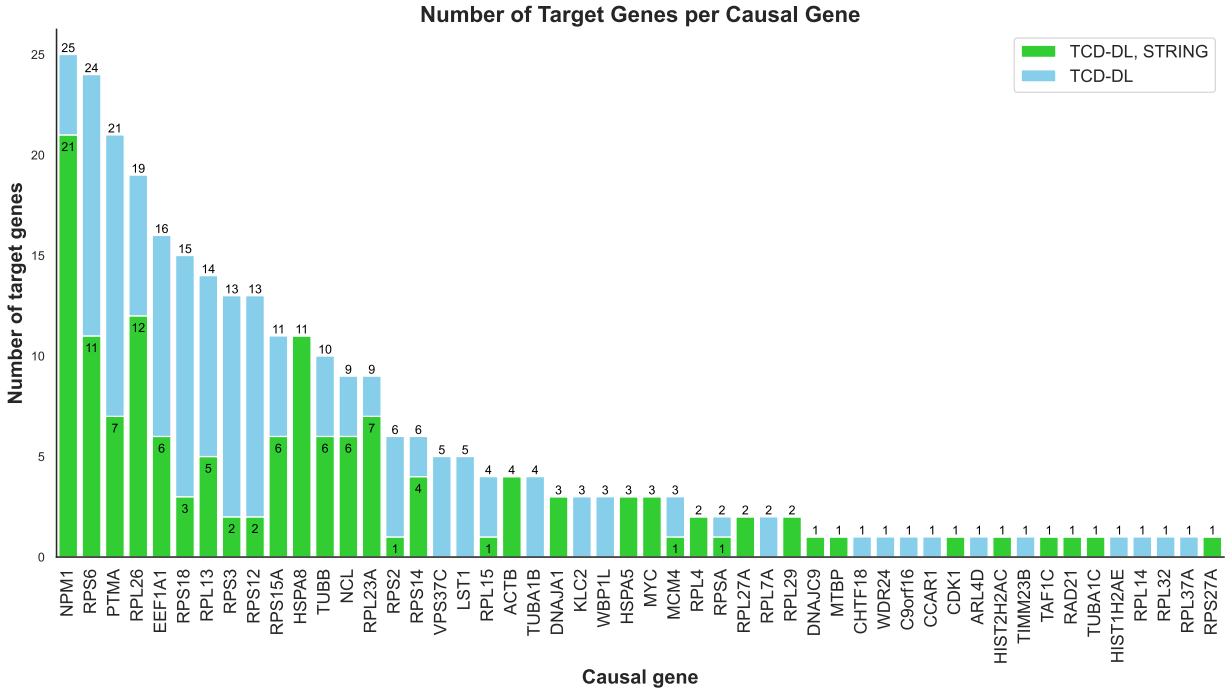


Figure 13: **Predicted influence of causal genes.** The histogram shows the number of leukemia-related target genes predicted to be regulated by each causal gene, with predictions made by TCD-DL (blue) and TCD-DL supported by STRING (green).

Table 9: **Benchmarking results.** Targeted cause discovery performance on E. coli GRN with 1565 genes over varying levels of simulator's observational fidelity. All measurements are expressed as percentages (%).

	Fidelity high			Fidelity medium			Fidelity low		
Method	AUROC	AP	F1	AUROC	AP	F1	AUROC	AP	F1
Random	50.0 ± 0.0	0.5 ± 0.0	0.5 ± 0.0	50.0 ± 0.0	0.5 ± 0.0	0.5 ± 0.0	50.0 ± 0.0	0.5 ± 0.0	0.5 ± 0.0
Correlation	51.1 ± 8.0	0.7 ± 0.2	0.8 ± 0.1	51.0 ± 7.6	0.7 ± 0.2	0.8 ± 0.1	49.8 ± 3.3	0.7 ± 0.1	0.8 ± 0.1
Sortnregress	49.8 ± 0.7	1.4 ± 0.3	1.3 ± 0.4	49.9 ± 0.6	1.4 ± 0.4	1.2 ± 0.4	50.1 ± 0.1	0.7 ± 0.1	0.7 ± 0.2
PMF-GRN	52.6 ± 2.3	0.9 ± 0.1	0.8 ± 0.4	52.9 ± 3.2	1.0 ± 0.2	1.0 ± 0.3	51.5 ± 2.1	1.0 ± 0.1	1.1 ± 0.3
GENIE3	56.0 ± 3.4	2.9 ± 0.8	2.1 ± 0.7	54.3 ± 3.9	2.2 ± 1.1	1.5 ± 0.9	52.2 ± 3.7	1.1 ± 0.2	0.6 ± 0.2
DCD-FG	50.0 ± 0.0	0.6 ± 0.0	1.0 ± 0.0	50.0 ± 0.0	0.6 ± 0.0	1.0 ± 0.0	50.0 ± 0.0	0.5 ± 0.1	1.0 ± 0.0
AVICI	56.5 ± 4.2	1.0 ± 0.2	0.4 ± 0.3	62.7 ± 5.5	3.1 ± 1.7	3.7 ± 2.6	56.5 ± 8.4	1.2 ± 0.9	1.3 ± 1.1
TCD-DL (ours)	**94.6** ± 1.8	**38.6** ± 6.3	**36.3** ± 6.1	**81.7** ± 11.3	**14.0** ± 11.7	**13.4** ± 12.0	**71.5** ± 3.7	**2.4** ± 1.5	**2.0** ± 1.8