DeepITE: Designing Variational Graph Autoencoders for Intervention Target Estimation

Hongyuan Tao[*]
Ant Group
Hangzhou, China
thy.qy@antgroup.com

Hang Yu[*]
Ant Group
Hangzhou, China
hyu.hugo@antgroup.com

Jianguo Li[†]
Ant Group
Hangzhou, China
lijg.zero@antgroup.com

Abstract

Intervention Target Estimation (ITE) is vital for both understanding and decision-making in complex systems, yet it remains underexplored. Current ITE methods are hampered by their inability to learn from distinct intervention instances collaboratively and to incorporate rich insights from labeled data, which leads to inefficiencies such as the need for re-estimation of intervention targets with minor data changes or alterations in causal graphs. In this paper, we propose DeepITE, an innovative deep learning framework designed around a variational graph autoencoder. DeepITE can concurrently learn from both unlabeled and labeled data with different intervention targets and causal graphs, harnessing correlated information in a self or semi-supervised manner. The model's inference capabilities allow for the immediate identification of intervention targets on unseen samples and novel causal graphs, circumventing the need for retraining. Our extensive testing confirms that DeepITE not only surpasses 13 baseline methods in the Recall@k metric but also demonstrates expeditious inference times, particularly on large graphs. Moreover, incorporating a modest fraction of labeled data (5-10%) substantially enhances DeepITE's performance, further solidifying its practical applicability. Our source code is available at `https://github.com/alipay/DeepITE`.

1 Introduction

Causal analysis in complex systems encompasses a series of steps beginning with causal discovery [1], which aims to delineate the causal structure, followed by the identification and estimation of causal effects [2]. Within this framework, a critical yet often overlooked component is Intervention Target Estimation (ITE) [3], alternatively known as Intervention Recognition [4]. ITE is the process of pinpointing which variables in a system have been subject to intervention, particularly when such interventions are opaque or not directly manipulable. This process not only fosters a deeper understanding of the causal mechanisms driving specific outcomes, which resonates with the principles of explainable AI (XAI), but also plays a pivotal role in recognizing variables that can be strategically altered to produce desired effects, aligning with the concept of algorithmic recourse [5].

To illustrate, consider the application of ITE in root cause analysis (RCA) within a microservices system. These systems consist of a network of services working in concert to deliver software functionality. When a service failure occurs, such as a system outage or performance degradation, it becomes imperative to identify the root cause. ITE is the key that unlocks definitive insight into the RCA process. It methodically pinpoints the specific services whose malfunctions—stemming from network issues, hardware failures, or security breaches—lead to the anomalies in question. ITE not

[*] Equal contribution.
[†] Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

only equips operators with a clear and logical explanation for the system's alerts, enhancing their comprehension of the issues at hand, but it also empowers them to implement immediate and effective countermeasures and fortify the system against future incidents. Beyond RCA in microservice systems, ITE's applicability extends to a multitude of domains, from unraveling the genetic factors involved in diseases within biomedicine, to tracing the determinants of user behavior for marketing.

Unfortunately, the field of ITE has not been thoroughly explored, often resulting in intervention targets being relegated to secondary outputs from causal discovery rather than being a dedicated field of inquiry. Only recently, Varici *et al.* [3, 6] pioneered the exclusive study of learning intervention targets in linear SCMs. In parallel, Li *et al.* [4] approached the problem from the perspective of RCA, coining it intervention recognition. Finally, Yang *et al.* [7] extended ITE to non-linear SCMs. This handful of methods can only identify the targets for an intervention instance[3] with a large sample size, relying on an accompanying dataset of known observational data, all within the confines of a fixed causal graph. The drawbacks of these strategies are twofold: **From the learning perspective**, they independently map data to intervention targets for each intervention instance, disregarding the potential correlations among distinct instances. In RCA scenarios, for example, various incidents could stem from the same underlying service problem. A collaborative learning approach, which considers all instances collectively, could more effectively elucidate the data-intervention target relationships. Moreover, these methods often neglect the labeled data that are often available, such as those obtained from controlled chaos engineering exercises that identify specific services as failure root causes. Consequently, opportunities to refine and expedite future similar analyses are lost. **From the inference standpoint**, slight changes in the data or in the graph structure necessitate a burdensome and complete re-estimation of intervention targets. In RCA contexts, this means that despite shared causality between distinct incidents, we still require piecemeal analyses for each new occurrence, leading to extended system downtime and delayed resolutions. Additionally, the premise that a large volume of data pertains to a uniform set of intervention targets is restrictive. Such data are challenging to obtain, which further complicates the assurance of these methods' performance.

Addressing these shortcomings, we introduce DeepITE, an innovative deep-learning solution that disentangles the learning and inference processes. In particular, **we design a variational graph autoencoder (VGAE) that can concurrently learn across diverse causal graphs and sets of intervention targets in a self-supervised or semi-supervised mode, thus, effectively harnesses correlated information to unravel the intricate relationship between input data and intervention targets. Once the VGAE is trained, its inference model can instantly identify intervention targets for new, unseen samples with different interventions and causal graphs, all without the need to retrain or refer to observational data.** Specifically, leveraging the principle that interventions entail the removal of all incoming edges to intervention targets, our VGAE framework is designed to estimate the probability of edge removal for each node, thereby identifying the intervention targets. The generative model within the VGAE employs a non-linear Graph Neural Network (GNN), an extension of linear SCMs that adheres to causal factorization and meets the criteria for causal interventions, across causal graphs of various structures and sizes. This theoretical foundation ensures robust ITE capabilities. The generative model accepts exogenous noise variables $\mathbf{u}$, the adjacency matrix of the observational causal graph $\boldsymbol{A}$, a Bernoulli-distributed intervention indicator γ characterizing edge removal probability, and outputs the distribution of endogenous variables $\mathbf{x}$. Conversely, the inference model interprets a given sample of endogenous variables $\mathbf{x}$ to infer distributions of exogenous variables $\mathbf{u}$ and the intervention indicator γ, all of which rely on the causal graph $\boldsymbol{A}$. We employ graph attention networks (GAT) as the backbone due to their flexibility and scalability. This VGAE—comprising both the generative and inference models—can be trained in a self-supervised manner by maximizing the evidence lower bound (ELBO) or can leverage labeled intervention targets when available for semi-supervised learning.

Our contributions can be summarized as follows:

- We propose a novel VGAE architecture tailored for ITE, termed DeepITE. It excels in collaborative learning from varying causal structures and interventions, negating the need for retraining with each new instance.
- We establish self-supervised and semi-supervised training approaches for DeepITE, allowing it to autonomously discern intervention targets and enhance accuracy through the integration of labeled data from controlled experiments.

[3] We define an intervention instance as one manipulation on the causal system. We can then collect a set of data with the same intervention targets for this instance.

- Extensive experiments show that DeepITE surpasses 13 baseline methods by a large margin on average in terms of *Recall@k* with competitive inference time, especially for large graphs.

2 Related Works

In this section, we briefly review the literature on ITE. Moreover, we notice that the realm of ITE is interconnected with causal explanations and RCA. These three concepts demonstrate a considerable degree of overlap (cf. [8, 4, 9]), suggesting that methods from each domain can not only inform and enhance one another but also be utilized in a complementary fashion. We therefore refer the readers to Appendix B for further discussion on causal explanations and RCA.

The limited literature on ITE approaches bifurcates, with one camp focusing on incidental estimation of intervention targets via causal discovery and the other dedicated solely to identifying intervention targets within a given causal framework. The former includes methods such as UT-IGSP [10], which seeks to recover an interventional Markov equivalence class (I-MEC) through permutation searches but is hampered by sample inefficiency and limited scalability. Ghassami *et al.* [11] explore linear structural causal models (SCMs) yet may struggle with complexity in diverse data settings. For causal insufficient systems, Jaber *et al.* [12] propose Ψ-FCI for matching interventional distributions to causal graphs and intervention target pairs, contending with exponential growth in complexity as the number of variables increases. Mooij *et al.* [13] alternatively propose a method leveraging context variables for integrating interventional datasets, but this method suffers from scalability issues with large graphs. To overcome this problem, RCD [9] further adapts the Ψ-FCI algorithm in [12] to the Ψ-PC algorithm to expedite the process in causally sufficient systems. The second approach, exemplified by CITE [3] and PreDITEr [6], zeros in on ITE by contrasting precision matrices from observational and interventional data, achieving scalability at the cost of being initially restricted to linear Gaussian SCMs. LIT [7] explores non-linear SCMs through non-linear ICA, still carrying quadratic complexity. Alternatively, CI-RCA [4] conducts ITE by detecting shifts in probability distributions of a variable conditioned on a variable's parents via hypothesis testing. Both groups of methods share critical disadvantages: they are vulnerable to even minor changes in data or causal graphs during both learning and inference, and they underutilize labeled data from controlled experiments.

3 Preliminaries

This section lays the groundwork for our study by introducing SCMs and Pearl's Causal Hierarchy.

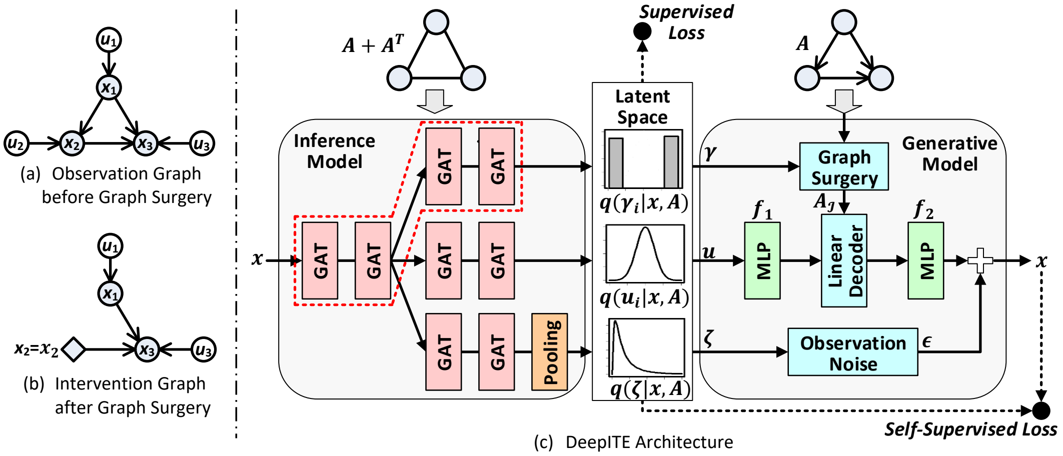
Structural Causal Models (SCMs): Given a set of variables $\mathbf{x} = [\mathrm{x}_1, \ldots, \mathrm{x}_d]$, SCMs present a formal mechanism to represent causal relations among them. An SCM is composed of two primary components: a set of structural equations and a causal graph. The structural equations take the form:

$$\mathrm{x}_i := f_i(\mathrm{Pa}(\mathrm{x}_i), \mathrm{u}_i), \quad i = 1, 2, \ldots, m, \tag{1}$$

where f_i is a deterministic function, $\mathrm{Pa}(\mathrm{x}_i)$ denotes parent variables that exert direct causal influence on x_i, and u_i signifies unobserved exogenous variables capturing influences not represented by other variables in $\mathbf{x}$. We typically invoke causal sufficiency, assuming the u_i are jointly independent, thereby ruling out hidden confounders. The use of the assignment symbol ":=" instead of an equality sign underscores the asymmetry of the causal relationship. The corresponding causal graph $\mathcal{G}$ (see Figure 1(a) for an example) induced by the SCM is typically a directed acyclic graph (DAG) with vertex set $\mathbf{x} \cup \mathbf{u}$ and directed edges from each variable on the right hand side (RHS) of a structural equation (1) to the variable on the left hand side, thus delineating the causal dependencies.

Pearl's Causal Hierarchy (PCH): Under the SCMs framework, PCH categorizes causal inference into a three-tiered structure reflective of the cognitive processes of "seeing" (observational), "doing" (interventional), and "imagining" (counterfactual). The initial tier addresses observational queries using SCMs as conventional probabilistic models to describe statistical associations.

Progressing to the second tier of PCH, SCMs distinguish themselves from standard probabilistic models by enabling the assessment of outcomes resulting from active interventions or manipulations, captured by the notions of *do-operator* and *graph surgery*. Here, the do-operator $\mathrm{do}(\mathrm{x}_i = x_i)$ represents an intervention that sets the variable x_i to value x_i, while graph surgery alters the corresponding causal graph by removing all incoming edges to the intervened-upon variable x_i. Interventional queries are then addressed by performing probabilistic inference in the modified graph, which often reveals new conditional independencies due to the excision of edges. For example, the interventional



(a) Observation Graph before Graph Surgery

(b) Intervention Graph after Graph Surgery

(c) DeepITE Architecture

Figure 1: **Left Panel**: Illustration of the do operator and the corresponding graph surgery: (a) The observation graph $\mathcal{G}$; (b) The intervention graph $\mathcal{G}_{\mathcal{I}}$ for $\mathrm{do}(\mathrm{x}_2 = x_2)$. **Right Panel** (c): The DeepITE architecture has an inference and a generative model. The inference model uses a three-branch GAT to link endogenous variables $\mathbf{x}$ to posterior distributions of intervention indicators γ_i, exogenous variables u_i, and observation noise precision ζ. The generative model then synthesizes $\mathbf{x}$ given these latent variables following Eq. (7) plus observation noise ϵ.

distribution $p(\mathrm{x}_3 | \, \mathrm{do}(\mathrm{x}_2 = x_2))$ for the SCM in Figure 1(b) is obtained via probabilistic inference with regard to (w.r.t.) the intervention graph:

$$p(\mathrm{x}_3 | \, \mathrm{do}(\mathrm{x}_2 = x_2)) = \sum_{\mathrm{x}_1} p(\mathrm{x}_1) p(\mathrm{x}_3 | \mathrm{x}_1, \mathrm{x}_2 = x_2), \tag{2}$$

contrasting with the conditional distribution in the original graph (i.e., Figure 1(a)):

$$p(\mathrm{x}_3 | \mathrm{x}_2 = x_2) = \sum_{\mathrm{x}_1} p(\mathrm{x}_1 | \mathrm{x}_2 = x_2) p(\mathrm{x}_3 | \mathrm{x}_1, \mathrm{x}_2 = x_2). \tag{3}$$

The key distinction is the marginal $p(\mathrm{x}_1)$ in (2) versus the conditional $p(\mathrm{x}_1 | \mathrm{x}_2 = x_2)$ in (3), reflecting that the causal relationship between x_1 and x_2 is broken by the intervention $\mathrm{do}(\mathrm{x}_2 = x_2)$.

Finally, there are *counterfactual* queries about what would or could have been, given that something else was in fact observed. We refer the readers to [14] as this topic is beyond the scope of our paper.

4 Problem Formulation

As discussed in the introduction, intervention target estimation (a.k.a. intervention recognition) is a pivotal component within the landscape of causal analysis, addressing the question of which nodes within a given causal system should be subjected to intervention in order to best explain the given interventional data. This inquiry draws upon the framework established by SCMs and operationalizes the principles enshrined in the second tier of Pearl's Causal Hierarchy. More formally,

Definition 1. *Given a causal graph $\mathcal{G}$ with variables $\mathbf{x}$, the observational data, and the interventional data corresponding to a certain intervention on a subset of variables $\mathbf{x}_{\mathcal{I}} \subseteq \mathbf{x}$, the task of intervention target estimation is to identify $\mathbf{x}_{\mathcal{I}}$.*

Here, we maintain the assumptions of causal sufficiency and the acyclicity of the causal graph. Note that while the DAG structure (i.e., the adjacency matrix A) is assumed to be known, the explicit forms of the structural equations remain unspecified. In comparison with the interventional queries mentioned in Section 3, which presuppose knowledge of where interventions have occurred and seek to determine their effects, ITE is the inverse process: it starts with the consequences of interventions and works backward to identify the sources of these perturbations. Essentially, we are solving for the *origin* of the observed interventional data, rather than predicting their *impact*.

In this paper, we innovatively solve the problem of intervention target estimation from the perspective of graph surgery. Recognizing that the interventional data align best with the intervention graph (see Figure 1(b)), our objective is to discover the subset of nodes $\mathbf{x}_{\mathcal{I}}$ such that, upon hypothetically removing the incoming edges to these nodes, the modified interventional model most accurately reflects the presented interventional data. Furthermore, We aim to create a singular model capable of pinpointing distinct sets of intervention targets for individual samples, each associated with its unique causal graph, while eliminating the need for observational data during inference. This represents a significant shift from conventional ITE methods that independently identify shared intervention targets for each intervention instance based on both observational and interventional data within the context of a fixed causal graph. Additionally, we seek to enhance model performance by incorporating labeling information during the training phase, which is a step forward in refining ITE processes.

5 DeepITE

To move forward to the above objectives, we present DeepITE, a VGAE that can estimate the probability of edge presence within the latent space. The architecture of DeepITE is showcased in Figure 1(c). We will first introduce the generative and inference models within the VGAE. Subsequently, we will explicate the self and semi-supervised methods to train the inference model.

5.1 Generative Model

The chief aim of the generative model is to recreate the observed variables $\mathbf{x}$ given the exogenous noise variables $\mathbf{u}$, the adjacency matrix $\boldsymbol{A}$ of the causal DAG, and a set of nodes $\mathcal{I}$ that have been intervened upon. Notably, when $\mathcal{I}$ is an empty set, the model is capable of recovering the observational distribution.

Specifically, when the structural equations are linear, they can be succinctly written as $\mathbf{x} = \boldsymbol{A}^T\mathbf{x} + \mathbf{u}$. Therefore, given $\mathbf{u}$ and $\boldsymbol{A}$, we can derive $\mathbf{x}$ through a linear decoder:

$$\mathbf{x} = (\boldsymbol{I} - \boldsymbol{A}^T)^{-1}\mathbf{u}. \tag{4}$$

Drawing inspiration from DAG-GNN [15], we extend this formulation to non-linear scenarios with:

$$\mathbf{x} = f_2\big((\boldsymbol{I} - \boldsymbol{A}^T)^{-1} f_1(\mathbf{u})\big), \tag{5}$$

where f_1 and f_2 are non-linear, component-wise learnable functions. In practice, these functions are executed by MLPs, which serve as universal approximators. Assuming f_2 is invertible, the aforementioned decoder corresponds to a conglomerate of non-linear structural equations [15]:

$$f_2^{-1}(\mathbf{x}) = \boldsymbol{A}^T f_2^{-1}(\mathbf{x}) + f_1(\mathbf{u}). \tag{6}$$

This setup implies that when f_1 and f_2^{-1} are suitably expressive, they can transform $\mathbf{u}$ and $\mathbf{x}$ into a space where their causal interrelations are aptly described by linear structural equations. The decoder, as specified in Eq. (5), exhibits inductiveness, enabling generalization to new nodes, edges, or graph schemas by only modifying the adjacency matrix $\boldsymbol{A}$ while preserving the learned functions f_1 and f_2. This decoder displays a particular characteristic, ratified by the following proposition:

Proposition 1. *For a GNN layer as defined in Eq.* (5)*, and denoting* $\mathrm{An}(i)$ *as the ancestor nodes of node i with the extension* $\mathrm{An}^*(i) = \mathrm{An}(i) \cup i$*, each output feature* x_i *exclusively acquires information from its own and all ancestor input features* $\mathbf{u}_{\mathrm{An}^*(i)}$.

Proof. See Appendix C. □

Owing to this property, this decoder (5) satisfies causal factorization and captures causal intervention, as proven below.

Proposition 2. *(causal factorization) The decoder defined in Eq.* (5) *conforms to causal factorization* $p(\mathrm{x}|\mathrm{u}, \boldsymbol{A}) = \prod_i p(\mathrm{x}_i|\mathbf{u}_{\mathrm{An}^*(i)})$*, that is, each endogenous variable* x_i *can be expressed as a function of its exogenous variable* u_i *and those of its causal ancestors.*

Proposition 3. *(causal intervention) The decoder defined in Eq.* (5) *captures causal interventions* $\mathrm{do}(\mathbf{x}_{\mathcal{I}} = x_{\mathcal{I}})$ *by replacing the original adjacency matrix* $\boldsymbol{A}$ *in Eq.* (5) *with the one corresponding to the post-intervention graph after graph surgery.*

Proof. See Appendices D and E. □

As established in Section 4, ITE resides within the second tier of PCH. Within this framework, any model purporting to tackle the ITE challenge must adeptly manage both observational and interventional data. The above propositions bridge this requirement, affirming DeepITE's competency in fulfilling the ITE task. Specifically, these propositions serve as the key to unlocking the model's ability to honor the causal structure inherent in the data and to emulate the effects of interventions.

In light of Proposition 3, to manipulate the intervened nodes $\mathcal{I}$ in the decoder (5), we introduce a Bernoulli distributed variable γ_i for each node x_i: $\gamma_i = 0$ means x_i is intervened, and thus, all incoming edges of x_i is removed during the graph surgery. The variable γ_i is henceforth referred to as the intervention indicator. The corresponding intervened adjacency matrix is given by $\boldsymbol{A}_{\mathcal{I}} = (\gamma^T \mathbf{1}) \odot \boldsymbol{A}$, where $\mathbf{1}$ is a column vector of all ones and $\odot$ denotes Hadamard product. As a result, the decoder, inclusive of interventions, is delineated as:

$$\mathbf{x} = \mathrm{Dec}(\mathbf{u}, \boldsymbol{\gamma}, \boldsymbol{A}) = f_2\Big(\big(\boldsymbol{I} - A_{\mathcal{I}}^T\big)^{-1} f_1(\mathbf{u})\Big). \tag{7}$$

When $\gamma_i = 1$ for all i, there is no intervention (i.e., $\boldsymbol{A}_{\mathcal{I}} = \boldsymbol{A}$) and the above decoder can describe the observational distribution.

To facilitate the reparameterization trick in VAE, we assume the exogenous variables $\mathbf{u}$ are standard normal distributions: $\mathrm{u}_i \sim \mathcal{N}(0, 1)$. Finally, since we do not have access to the true structural equations, we introduce the observation noise $\epsilon \sim \mathcal{N}(0, \zeta^{-1})$ to (7), so as to account for the model uncertainty associated with the above decoder (7). Here, ζ denotes the inverse variance of the noise, and we impose a non-informative Jeffrey's prior on ζ, that is, $p(\zeta) \propto 1/\zeta$.

Collectively, the overall generative model can be factorized as:

$$p(\mathbf{x}, \mathbf{u}, \boldsymbol{\gamma}, \zeta | \boldsymbol{A}) = p(\mathbf{x} | \mathbf{u}, \boldsymbol{\gamma}, \boldsymbol{A}, \zeta) p(\zeta) \prod_{i=1}^{m} p(\mathrm{u}_i) p(\gamma_i), \tag{8}$$

where

$$\begin{aligned} &p(\mathrm{u}_i) = \mathcal{N}(0, 1), \quad p(\gamma_i) = \mathrm{Bern}(\pi) \quad \forall i, \\ &p(\zeta) \propto 1/\zeta, \quad p(\mathbf{x} | \mathbf{u}, \boldsymbol{\gamma}, \boldsymbol{A}, \zeta) = \mathcal{N}\big(\mathrm{Dec}(\mathbf{u}, \boldsymbol{\gamma}, \boldsymbol{A}), \zeta^{-1} \boldsymbol{I}\big). \end{aligned} \tag{9}$$

Here, π denotes the probability of taking 1 in a Bernoulli distribution and $\boldsymbol{I}$ is the identity matrix.

Discussion on Hard versus Soft Interventions: Hard interventions, characterized by the removal of incoming edges as part of graph surgery, contrast with soft interventions, which modify the causal mechanism without complete elimination. For instance, for an intervened node x_i, a soft intervention would replace the original structural equation $\mathrm{x}_i := f_i(\mathrm{Pa}(\mathrm{x}_i), \mathrm{u}_i)$ with an updated version $\mathrm{x}_i := f_i'(\mathrm{Pa}(\mathrm{x}_i), \mathrm{u}_i)$, where $f_i \neq f_i'$, thereby altering the generative process while maintaining the graph's structure. In our generative model, the intervention indicator γ_i is a Bernoulli variable, allowing the learning of edge removal probability from data $\mathbf{x}$. A probability of $\gamma_i = 0$ being one indicates a hard intervention, whereas any other value suggests a soft intervention. This nuanced approach allows DeepITE to offer a spectrum between hard and soft interventions based on given data.

5.2 Inference Model

The pinnacle goal of the inference model within DeepITE is to determine the probability that a node i has undergone an intervention based on the observed data, succinctly expressed as $p(\gamma_i = 1 | \mathbf{x})$. To achieve this, we aim to compute the exact posterior $p(\mathbf{u}, \boldsymbol{\gamma}, \zeta | \mathbf{x}, \boldsymbol{A})$. However, the intricate nature of this posterior necessitates approximation through a tractable inference model $q(\mathbf{u}, \boldsymbol{\gamma}, \zeta | \mathbf{x}, \boldsymbol{A})$ [16, 17]. Specifically, the inference model can be factorized in a manner akin to the generative model as:

$$q(\mathbf{u}, \boldsymbol{\gamma}, \zeta | \mathbf{x}, \boldsymbol{A}) = q(\zeta | \mathbf{x}, \boldsymbol{A}) \prod_{i=1}^{m} q(\mathrm{u}_i | \mathbf{x}, \boldsymbol{A}) q(\gamma_i | \mathbf{x}, \boldsymbol{A}). \tag{10}$$

The variational distributions on RHS are parametrized by:

$$q(\mathrm{u}_i | \mathbf{x}, \boldsymbol{A}) = \mathcal{N}\big(\mu_i(\mathbf{x}, \boldsymbol{A}), \sigma_i^2(\mathbf{x}, \boldsymbol{A})\big), \tag{11}$$

$$q(\gamma_i | \mathbf{x}, \boldsymbol{A}) = \mathrm{Bern}\,\big(\omega_i(\mathbf{x}, \boldsymbol{A})\big), \tag{12}$$

$$q(\zeta | \mathbf{x}, \boldsymbol{A}) = \mathrm{Lognormal}\,\big(\mu_\zeta(\mathbf{x}, \boldsymbol{A}), \sigma_\zeta^2(\mathbf{x}, \boldsymbol{A})\big), \tag{13}$$

where the Bernoulli distribution (26) can be well approximated using the Gumbel-Softmax reparameterization trick [18, 19]. The parameters of the variational q distributions in (25)-(27) are derived from a network based on Graph Attention Networks (GAT). While any inductive spatial GNN can be used as the inference network in DeepITE, we choose GAT since it provides the flexibility and scalability necessary for our model. This flexibility stems from GAT's ability to dynamically weigh the importance of different nodes, thus allowing the variational distribution given by the inference network better approximate the exact posterior distribution. This advantage is further demonstrated in Appendix G.6, where we replace the GAT encoder with the encoder of DAG-GNN.

The architecture features an initial dual-layer GAT for feature extraction, followed by three specialized branches dedicated to the parameters of $\mathbf{u}$, $\boldsymbol{\gamma}$, and ζ. Note that the parameters of $\mathbf{u}$ and $\boldsymbol{\gamma}$ can be regarded as node-level features, while those of ζ as graph-level features. As such, the final ζ branch includes a pooling layer to yield the graph-level features. The complete inference network is depicted in Figure 1(c).

It is pertinent to mention that the graph associated with the GATs is undirected, in contrast to the directed nature of the causal graph. This design choice is motivated by the need for the variational update of a variable to account for all elements within its Markov blanket, which includes the parents, children, and co-parents of the node [20]. An undirected graph facilitates the message passing process within this Markov blanket by removing the constraints imposed by edge directionality. Moreover, the use of GAT ensures that the resulting model is inductive, enabling its application to new nodes within the graph and even entirely new graph structures.

Relation to DAG-GNN [15]: DAG-GNN's objective is to infer the structure of a DAG (i.e., the zero pattern of $\boldsymbol{A}$) from the provided data, a process known as causal discovery. The architecture of DAG-GNN employs a generative model expressed as $\mathbf{x} = f_2((\boldsymbol{I} - \boldsymbol{A}^T)^{-1} f_1(\mathbf{u}))$ and an inference model as $\mathbf{u} = f_4((\boldsymbol{I} - \boldsymbol{A}^T) f_3(\mathbf{x}))$, both of which are differentiable with respect to $\boldsymbol{A}$. This differentiability is crucial as it enables the learning of $\boldsymbol{A}$ through gradient descent. On the other hand, DeepITE enhances the generative model by integrating an intervention indicator, which facilitates the adaptation of the model to account for interventions via graph surgery. Furthermore, DeepITE's inference model seeks to closely approximate the true posterior $p(\mathbf{u}, \boldsymbol{\gamma}, \zeta | \mathbf{x}, \boldsymbol{A})$. Unlike DAG-GNN, which may only collect messages from the parents of a node, DeepITE's model is designed to aggregate messages from all nodes within the Markov blanket of a given node i. This comprehensive approach ensures that DeepITE's inference model is not as restrictive as DAG-GNN's and is better suited for ITE tasks.

Relation to VACA [21]: VACA sets out to perform causal queries utilizing observational data within the framework of the VGAE. It hinges on Message Passing Neural Networks (MPNNs) for both the generative and inference models. A prerequisite for VACA to perform observational and interventional queries is that the generative model's number of MPNN layers must at least be $\delta - 1$ given that δ is the graph diameter.[4] This criterion ensures that the information propagated within the graph can reach from one end to the other, thereby reflecting the global structure necessary for accurate causal inference. DeepITE aligns with this requirement for effectively performing ITE. However, DeepITE distinguishes itself by employing the generative model, specified in Eq. (7). It keeps the number of GNN layers to be one regardless of graph diameter, while satisfying causal factorization and intervention conditions (cf. Proposition 2-3). DeepITE thereby overcomes the limitations imposed by VACA's dependence on graph diameter, offering a substantial benefit for collective learning on graphs with different sizes.

5.3 Self and Semi-Supervised Learning

DeepITE's learning strategies encompass self-supervised learning, which automates the identification of intervention targets from unlabeled data, and semi-supervised learning, which refines the model's performance by integrating labeled data. The training process is summarized in Algorithm 1.

Self-Supervised Learning: Given the generative and inference model, we can learn their parameters jointly by maximizing the evidence lower bound (ELBO) of the log-likelihood of the given data $\mathbf{x}$:

$$\mathcal{L} = \mathbb{E}_q[\log p(\mathbf{x}, \mathbf{u}, \boldsymbol{\gamma}, \zeta | \boldsymbol{A})] + \mathbb{H}_q \leq \log p(\mathbf{x} | \boldsymbol{A}), \tag{14}$$

where $\mathbb{E}_q$ denotes expectation over the q distribution in (10) and $\mathbb{H}_q$ denotes the entropy of the q distribution. The derivation of the ELBO can be found in Appendix F. Note that $\mathcal{L}$ can be maximized via stochastic gradient ascent after using the reparameterization trick for normal and Bernoulli distributions [22, 23, 19].

Semi-Supervised Learning: Information regarding intervention targets may often be available in practice. For instance, in the case of RCA in cloud-native systems, the ground truth of intervention targets can be derived from resolved incidents and chaos engineering exercises. This ground truth data can be utilized to train the inference network, enabling it to more accurately identify intervention targets. In particular, the term $q(\gamma_i | \mathbf{x}, \boldsymbol{A})$ is replaced by the ground truth γ_i^* when computing the ELBO $\mathcal{L}$, and an additional term is introduced to maximize the log-likelihood $\log q(\gamma_i^* | \mathbf{x}, \boldsymbol{A})$. By taking advantage of both labeled and unlabeled data, we can effectively train the inference model.

Once trained, the inference model of DeepITE becomes equipped to evaluate individual new samples against different causal graphs, directly deducing the intervention targets and thus circumventing the necessity of retraining for each new scenario. It can also distinguish between observational and interventional data directly as the former equates to the absence of intervention targets. Another significant feature of DeepITE is its independence from observational data during the testing phase; it relies solely on the interventional data input into the inference network. This is a distinct advantage over existing ITE methods, which consistently require observational data for ITE. In practice, we are primarily concerned with the intervention indicators $\boldsymbol{\gamma}$. Hence, during inference, we only need

[4] The diameter of a graph is the length of the shortest path between the most distanced endogenous nodes.

Table 1: Recall@k of different algorithms for detecting the intervened nodes from the Synthetic dataset. Graph-m means DAGs with m nodes.

DATASET	**Graph-50**		**Graph-100**		**Graph-500**	
METRICS	Recall@1	Recall@5	Recall@1	Recall@5	Recall@1	Recall@5
UT-IGSP	0.224 ± 0.015	0.318 ± 0.019	0.079 ± 0.007	0.185 ± 0.010	0.016* ± 0.002	0.020* ± 0.004
CITE	0.098 ± 0.009	0.124 ± 0.013	0.044 ± 0.003	0.063 ± 0.006	0.007 ± 0.001	0.008 ± 0.001
PreDITEr	0.104 ± 0.009	0.122 ± 0.012	0.049 ± 0.004	0.066 ± 0.006	0.008 ± 0.001	0.008 ± 0.001
TreeExplainer	0.381 ± 0.022	0.510 ± 0.017	0.298 ± 0.016	0.448 ± 0.009	0.102 ± 0.008	0.152 ± 0.002
ASV	0.296 ± 0.021	0.390 ± 0.022	0.261 ± 0.014	0.323 ± 0.017	0.081 ± 0.003	0.140 ± 0.005
ShapleyFlow	0.552 ± 0.017	0.690 ± 0.009	0.378 ± 0.009	0.485 ± 0.007	0.124 ± 0.005	0.148 ± 0.002
PWSHAP	0.468 ± 0.014	0.610 ± 0.012	0.339 ± 0.009	0.454 ± 0.008	0.117 ± 0.003	0.195 ± 0.003
CauseInfer	0.561 ± 0.002	0.765 ± 0.003	0.554 ± 0.002	0.786 ± 0.02	0.559 ± 0.003	0.769 ± 0.004
MicroHECL	0.462 ± 0.010	0.587 ± 0.009	0.341 ± 0.004	0.400 ± 0.004	0.199 ± 0.003	0.241 ± 0.004
MicroRCA	0.647 ± 0.004	0.899 ± 0.003	0.623 ± 0.004	0.875 ± 0.004	0.436 ± 0.003	0.676 ± 0.003
CausalRCA	0.633 ± 0.004	0.894 ± 0.004	0.622 ± 0.004	0.863 ± 0.004	0.418 ± 0.004	0.630 ± 0.004
CI-RCA	0.615 ± 0.002	0.952 ± 0.001	0.631 ± 0.002	0.930 ± 0.003	0.623 ± 0.004	0.823 ± 0.003
RCD	0.495 ± 0.003	0.706 ± 0.004	0.440 ± 0.004	0.521 ± 0.005	0.325 ± 0.002	0.364 ± 0.003
DeepITE (sep)	**0.723** ± 0.002	**0.972** ± 0.003	0.685 ± 0.004	**0.968** ± 0.002	**0.642** ± 0.003	**0.891** ± 0.004
DeepITE (mix)	0.718 ± 0.003	0.945 ± 0.005	**0.690** ± 0.004	0.923 ± 0.003	0.627 ± 0.003	0.875 ± 0.004

to process $\mathbf{x}$ through the relevant branch of $\mathbf{x}$, as highlighted by the red dashed box in Figure 1(c), disregarding the other branches to optimize inference efficiency.

6 Experimental Results

In this section, we demonstrate the usefulness of DeepITE on three datasets, comprising one synthetically generated dataset, which provides a controlled environment to test the robustness and scalability of the framework, and two real-world datasets that introduce the complexity of genuine causal systems. We position DeepITE against 13 state-of-the-art (SOTA) methods, spanning three areas of relevance: Intervention Target Estimation (ITE), Explainable AI (XAI), and Root Cause Analysis (RCA), due to their intertwined nature (see more discussions in Section 2 and Appendix B).

- **ITE**: We select 3 methods: UT-IGSP [10], which learns intervention targets as a byproduct of causal discovery; CITE [3] and PreDITEr [6], both of which are dedicated to ITE.
- **XAI**: We opt for TreeExplainer [24], ASV [25], ShapleyFlow [26], and PWSHAP [27], 4 methods based on Shapley values. TreeExplainer only considers associations, whereas ASV, ShapleyFlow, and PWSHAP incorporate causation, accounting for the DAG structure.
- **RCA**: We pick 6 methods: CauseInfer [28], MicroHECL [29], MicroRCA [30], CausalRCA [31], CI-RCA [4], and RCD [9]. The last two aim to find intervention targets in a causal graph.

To facilitate a fair comparison, all methods are provided with the same ground truth causal graph, eschewing the need for graph construction from data for some RCA methods. More implementation details can be found in Appendix G.1. The performance is quantified using the *Recall@k* metric. *Recall@k* measures the proportion of true intervention targets (ITs) that are successfully captured within the top k ranked candidates proposed by each method. This metric is widely adopted in the literature [4, 9, 29]. When $k = 1$, our goal is to pinpoint the intervention targets based on the highest-ranked candidate. We prioritize *Recall@k* because, in practice, false positives can be eliminated through further analysis, while false negatives are irrecoverable as they get lost among the numerous true negatives. All experiments report average results over 10 trials, with error bars representing a standard deviation ($\pm 1\sigma$) from the mean.

Synthetic Data: The synthetic data is generated following the method outlined in CI-RCA [4] and in Appendix G.2. We assess causal graphs with nodes ranging from 50 to 500 and corresponding edges from 100 to 5000, exhibiting different levels of complexity. In particular, DeepITE is evaluated in two configurations: DeepITE (sep), where separate models are trained for each graph size, and DeepITE (mix), where a single model is trained across all graph sizes and structures.

Analysis of the results in Table 1 reveals 5 key insights: (i) **Traditional ITE Methods Fall Short**: Such methods underperform with larger graphs with fewer samples due to their design for small, dense datasets (typically involving tens of nodes but with tens of thousands of samples) and lack of cross-instance learning, treating each intervention instance in isolation. As shown in Appendix G.3, they perform much better for small graphs with large sample size. (ii) **XAI Methods Face Challenges**: TreeExplainer lacks consideration for causal relationships in graphs. ASV and Shapley Flow, although

Table 2: Results of the Protein Signaling Data and the ICASSP-SPGC Data.

DATASET	**Protein Signaling**	**ICASSP-SPGC 2022**			
METRICS	Recall@1	Recall@1	Recall@5	Root.Acc	Score
UT-IGSP	0.579 ± 0.018	-	-	-	-
CITE	0.588 ± 0.011	-	-	-	-
PreDITEr	0.586 ± 0.010	-	-	-	-
TreeExplainer	0.434 ± 0.020	0.367 ± 0.013	0.687 ± 0.010	0.7401 ± 0.0153	0.3534 ± 0.0298
ASV	0.441 ± 0.017	0.449 ± 0.010	0.720 ± 0.008	0.7933 ± 0.0117	0.3820 ± 0.0230
ShapleyFlow	0.615 ± 0.021	0.677 ± 0.013	0.825 ± 0.015	0.9176 ± 0.0131	0.5312 ± 0.0252
PWSHAP	0.603 ± 0.016	0.488 ± 0.009	0.741 ± 0.010	0.8551 ± 0.0109	0.4233 ± 0.0211
CauseInfer	0.076 ± 0.002	0.278 ± 0.003	0.490 ± 0.001	0.5808 ± 0.0084	0.1139 ± 0.0180
MicroHECL	0.081 ± 0.004	0.323 ± 0.010	0.661 ± 0.011	0.7339 ± 0.0134	0.3697 ± 0.0283
MicroRCA	0.127 ± 0.003	0.246 ± 0.004	0.463 ± 0.002	0.5662 ± 0.0089	0.0721 ± 0.0204
CausalRCA	0.113 ± 0.001	0.308 ± 0.004	0.447 ± 0.003	0.5353 ± 0.0078	0.0617 ± 0.0161
CI-RCA	0.090 ± 0.001	0.559 ± 0.002	0.828 ± 0.001	0.9284 ± 0.0055	0.5650 ± 0.0105
RCD	0.214 ± 0.002	0.481 ± 0.008	0.757 ± 0.005	0.8768 ± 0.0112	0.4542 ± 0.0216
DeepITE	**0.652** ± 0.002	**0.881** ± 0.002	**0.984** ± 0.000	**0.9794** ± 0.0023	**0.9085** ± 0.0040

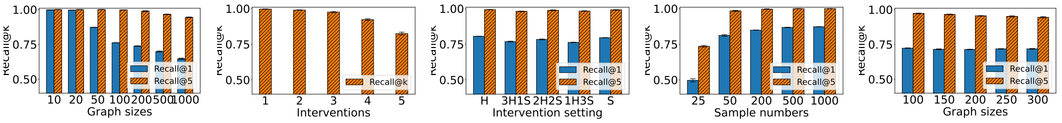
aware of the causal structure, struggle with scalability similar to traditional ITE methods, as these methods demonstrate optimal performance on comparatively smaller graphs (around 10-20 nodes) (cf. [25, 26]). (iii) **RCA Methods Show Promise but Have Limitations**: RCA methods generally perform better than ITE and XAI approaches, with CI-RCA aligning exactly with the ITE task. However, these methods also face challenges in integrating multiple intervention instances effectively. (iv) **DeepITE Models Excel**: Both DeepITE (sep) and DeepITE (mix) outperform all benchmarked methods on Recall@1 and Recall@5 metrics, attributing success to a flexible model that fosters collaborative instance learning, independent of graph characteristics, enabling precise alignment of data with intervention targets. (v) **Competitive Performance within DeepITE Models**: The two DeepITE models (sep&mix) demonstrate competitive results, indicating that the model's inductive strength and its adaptability to various graph structures and sizes.

Protein Signaling Dataset: The description of the dataset is presented in Appendix G.4. The results, shown in the second column of Table 2, focus on the Recall@1 metric due to the graph's limited size. It is evident that DeepITE outshines competing methods, attributable to its capacity for collaborative learning across the entire dataset and its inherent adaptability. Traditional ITE and XAI methods trail closely behind, with their methodologies being particularly suited to smaller graphs that nevertheless have a substantial number of samples. In contrast, RCA methods exhibit weaker performance as techniques such as PageRank, DFS, BFS, or rank walk struggle to distinguish between nodes in such a compact network. Unlike these approaches, DeepITE demonstrates versatility in handling both small and large graphs, affirming its utility across a wide range of practical scenarios.

ICASSP-SPGC 2022: The details of this dataset can be found in Appendix G.4. Note that the absence of purely observational data in this dataset precludes the application of the ITE methods including UT-IGSP, CITE, and PreDITEr. For our evaluation, we continue to employ Recall@1 and Recall@5 metrics and additionally consider accuracy and the root-cause score, the latter two being an official recommendation. The root-cause score is calculated by subtracting the number of false positives from the number of true positives and then dividing by the total number of true intervention targets. The results of this multifaceted assessment are presented in the last four columns of Table 2. Once more, in comparison with the SOTA methods, DeepITE stands out by a large margin of above 20% in Recall@1, underscoring its applicability to real-world RCA challenges. Notably, CI-RCA and Shapley Flow emerge as close second-best performers, likely because both methods leverage causal rather than just correlational information, which appears to be advantageous in this context.

6.1 Ablation Study

Due to the page limit, we only present an overview of the major findings here. More details can be found in Appendix G.5- G.7. (i) **Impact of Label Proportions**: Incorporating even a modest amount (5-10%) of labeled data significantly enhances DeepITE's performance across various datasets, with recall improvements ranging from 4-20%. This approach provides substantial benefits in practical settings, unlike other baseline methods in the study that cannot utilize labeled data at all. (ii) **Replacement of Encoder and Decoder**: Modifying DeepITE's encoder to DAG-GNN's and its decoder to VACA's in two ablation designs shows that DeepITE–with its flexible generative and inference models–outperforms both modified versions and the original VACA [21] and DAG-GNN [15]. These observations, especially under conditions of increasing graph complexity, highlight



(a) Graph sizes (b) Interventions (c) Mixed hard&soft (d) Graph samples (e) Mixed graphs

Figure 2: The performance of DeepITE as a function of (a) graph sizes, (b) interventions, (c) the mixture proportion of soft and hard interventions, (d) sample size for each graph, (e) the number of mixed graphs.

DeepITE's robustness and the limitations of DAG-GNN's rigid design and VACA's requirement of minimal decoder layers in estimating ITE. (iii) **Scalability**: Figure 2(a-b) revealed the performance of DeepITE as a function of the graph size and the number of interventions respectively. Our findings indicate that while performance in terms of Recall@1 declines as the graph size increases, Recall@5 remains stable even for graphs with 1000 nodes—a size that is already considered quite large for causal analysis. (iv) **Hard&Soft Intervention**: We examined DeepITE under varying ratios of hard and soft interventions, where soft interventions were modeled by modifying the linear structural equations of the intervention targets to quadratic forms. Figure 2(c) shows DeepITE's adaptability to these mixtures, confirming its effectiveness in handling both types of interventions with only minimal performance loss. (v) **Samples**: Figure 2(d) illustrates the performance of DeepITE with variations in sample size. Our results show that DeepITE exhibits robustness with performance generally improving as the sample size increases. In contrast, traditional ITE methods [3, 6, 10] typically require thousands of samples for a single graph and intervention set to perform well. This resilience can be attributed to the collaborative learning framework of DeepITE and the relatively few parameters in the GNN-based encoder and decoder. (vi) **Amortization**: The performance of DeepITE as more graphs with different sizes are trained together, is detailed in Figure 2(e). Our findings indicate a minimal gradual degradation in the performance of DeepITE (mix) as we incorporate more graphs of varying sizes, attributed to the amortization error. Moreover, Table 1 shows that DeepITE (mix) even outperforms DeepITE (sep) training exclusively on 100-node graphs in terms of Recall@1. Based on this evidence, we maintain that the amortization process across graphs does not significantly hinder the performance.

6.2 Runtime Analysis

We conducted a runtime analysis using four distinct datasets, with variable counts m ranging from 5 to 500, specifically $m = [5, 10, 11, 20, 50, 100, 500]$. For each value of m, we executed 10 trials on a set of 1000 samples and reported the average runtime. To ensure a fair comparison, we focused exclusively on the code pertinent to intervention identification. Timing commenced the moment the algorithm received the dataset and accompanying causal graph, if applicable, and ceased immediately upon delivery of the final results. This process ensured that our analysis exclusively measured the performance of the algorithm's core intervention-targeting functionality.

The runtime performance of the various methods, relative to graph size, is depicted in Appendix Figure 3. From the analysis, we note that DeepITE's runtime curve, represented in black, has the gentlest slope, implying that it boasts the lowest time complexity among all the methods. Notably, DeepITE secures the shortest runtime for graphs with more than 100 nodes. This heightened efficiency is attributable to DeepITE's inference process, which necessitates only a single pass through one branch of the inference network. In contrast, UT-IGSP exhibits the highest time complexity as it engages in an exponentially growing number of hypothesis tests to identify intervention targets. For instance, when handling graphs with $m = 500$ nodes, UT-IGSP requires nearly an hour to complete a single run.

7 Conclusion

In this paper, we presented DeepITE, a novel VGAE for ITE. By carefully design the VGAE based on GNNs, DeepITE allows collaborative learning and amortized inference across data with a range of intervention targets and causal graphs. The model adeptly supports both self-supervised and semi-supervised learning modalities, effectively harnessing labeled data to refine ITE. Our comprehensive results demonstrate that DeepITE can be seamlessly adapted to a multitude of domains, accommodating diverse causal graph configurations while exhibiting superior performance in terms of both Recall@k metrics and computational efficiency.

Acknowledgements

We would like to thank Ant Group for their support for this work.

References

[1] Judea Pearl. Causal inference in statistics: An overview. 2009.

[2] Judea Pearl. *Causality*. Cambridge university press, 2009.

[3] Burak Varici, Karthikeyan Shanmugam, Prasanna Sattigeri, and Ali Tajer. Scalable intervention target estimation in linear models. *Advances in Neural Information Processing Systems*, 34:1494–1505, 2021.

[4] Mingjie Li, Zeyan Li, Kanglin Yin, Xiaohui Nie, Wenchi Zhang, Kaixin Sui, and Dan Pei. Causal inference-based root cause analysis for online service systems with intervention recognition. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3230–3240, 2022.

[5] Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. A survey of algorithmic recourse: contrastive explanations and consequential recommendations. *ACM Computing Surveys*, 55(5):1–29, 2022.

[6] Burak Varici, Karthikeyan Shanmugam, Prasanna Sattigeri, and Ali Tajer. Intervention target estimation in the presence of latent variables. In James Cussens and Kun Zhang, editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 2013–2023. PMLR, 01–05 Aug 2022.

[7] Yuqin Yang, Saber Salehkaleybar, and Negar Kiyavash. Learning unknown intervention targets in structural causal models from heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pages 3187–3195. PMLR, 2024.

[8] Chaoyu Chen, Hang Yu, Zhichao Lei, Jianguo Li, Shaokang Ren, Tingkai Zhang, Silin Hu, Jianchao Wang, and Wenhui Shi. Balance: Bayesian linear attribution for root cause localization. *Proceedings of the ACM on Management of Data*, 1(1):1–26, 2023.

[9] Azam Ikram, Sarthak Chakraborty, Subrata Mitra, Shiv Saini, Saurabh Bagchi, and Murat Kocaoglu. Root cause analysis of failures in microservices through causal discovery. *Advances in Neural Information Processing Systems*, 35:31158–31170, 2022.

[10] Chandler Squires, Yuhao Wang, and Caroline Uhler. Permutation-based causal structure learning with unknown intervention targets. In *Conference on Uncertainty in Artificial Intelligence*, pages 1039–1048. PMLR, 2020.

[11] AmirEmad Ghassami, Saber Salehkaleybar, Negar Kiyavash, and Kun Zhang. Learning causal structures using regression invariance. *Advances in Neural Information Processing Systems*, 30, 2017.

[12] Amin Jaber, Murat Kocaoglu, Karthikeyan Shanmugam, and Elias Bareinboim. Causal discovery from soft interventions with unknown targets: Characterization and learning. *Advances in neural information processing systems*, 33:9551–9561, 2020.

[13] Joris M Mooij, Sara Magliacane, and Tom Claassen. Joint causal inference from multiple contexts. *Journal of Machine Learning Research*, 21(99):1–108, 2020.

[14] Judea Pearl. The structural theory of counterfactuals. In *Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems (NIPS 2011)*, pages 2399–2407, 2011.

[15] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. Dag-gnn: Dag structure learning with graph neural networks. In *International Conference on Machine Learning*, pages 7154–7163. PMLR, 2019.

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[17] Samuel Gershman and Noah Goodman. Amortized inference in probabilistic reasoning. In *Proceedings of the annual meeting of the cognitive science society*, volume 36, 2014.

[18] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations (ICLR 2017)*. OpenReview. net, 2017.

[19] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

[20] John Winn, Christopher M Bishop, and Tommi Jaakkola. Variational message passing. *Journal of Machine Learning Research*, 6(4), 2005.

[21] Pablo Sánchez-Martin, Miriam Rateike, and Isabel Valera. Vaca: Designing variational graph autoencoders for causal queries. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8159–8168, 2022.

[22] Diederik P Kingma and Max Welling. Stochastic gradient vb and the variational auto-encoder. In *International Conference on Learning Representations*, volume 19, page 121, 2014.

[23] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

[24] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[25] Christopher Frye, Colin Rowat, and Ilya Feige. Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability. *Advances in Neural Information Processing Systems*, 33:1229–1239, 2020.

[26] Jiaxuan Wang, Jenna Wiens, and Scott Lundberg. Shapley flow: A graph-based approach to interpreting model predictions. In *International Conference on Artificial Intelligence and Statistics*, pages 721–729. PMLR, 2021.

[27] Lucile Ter-Minassian, Oscar Clivio, Karla Diazordaz, Robin J Evans, and Christopher C Holmes. Pwshap: a path-wise explanation model for targeted variables. In *International Conference on Machine Learning*, pages 34054–34089. PMLR, 2023.

[28] Pengfei Chen, Yong Qi, Pengfei Zheng, and Di Hou. Causeinfer: Automatic and distributed performance diagnosis with hierarchical causality graph in large distributed systems. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pages 1887–1895. IEEE, 2014.

[29] Dewei Liu, Chuan He, Xin Peng, Fan Lin, Chenxi Zhang, Shengfang Gong, Ziang Li, Jiayu Ou, and Zheshun Wu. Microhecl: High-efficient root cause localization in large-scale microservice systems. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pages 338–347. IEEE, 2021.

[30] Li Wu, Johan Tordsson, Erik Elmroth, and Odej Kao. Microrca: Root cause localization of performance issues in microservices. In *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9. IEEE, 2020.

[31] Ruyue Xin, Peng Chen, and Zhiming Zhao. Causalrca: Causal inference based precise fine-grained root cause localization for microservice applications. *Journal of Systems and Software*, page 111724, 2023.

[32] Patrick Schwab and Walter Karlen. Cxplain: Causal explanations for model interpretation under uncertainty. *Advances in neural information processing systems*, 32, 2019.

[33] Wanyu Lin, Hao Lan, and Baochun Li. Generative causal explanations for graph neural networks. In *International Conference on Machine Learning*, pages 6666–6679. PMLR, 2021.

[34] Sahil Verma, John Dickerson, and Keegan Hines. Counterfactual explanations for machine learning: Challenges revisited. *arXiv preprint arXiv:2106.07756*, 2021.

[35] Amir-Hossein Karimi, Julius Von Kügelgen, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. *Advances in neural information processing systems*, 33:265–277, 2020.

[36] Amir-Hossein Karimi, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse: from counterfactual explanations to interventions. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 353–362, 2021.

[37] Zijie Guan, Jinjin Lin, and Pengfei Chen. On anomaly detection and root cause analysis of microservice systems. In *Service-Oriented Computing–ICSOC 2018 Workshops: ADMS, ASOCA, ISYyCC, CloTS, DDBS, and NLS4IoT, Hangzhou, China, November 12–15, 2018, Revised Selected Papers 16*, pages 465–469. Springer, 2019.

[38] JinJin Lin, Pengfei Chen, and Zibin Zheng. Microscope: Pinpoint performance issues with causal graphs in micro-service environments. In *Service-Oriented Computing: 16th International Conference, ICSOC 2018, Hangzhou, China, November 12-15, 2018, Proceedings 16*, pages 3–20. Springer, 2018.

[39] Weilan Lin, Meng Ma, Disheng Pan, and Ping Wang. Facgraph: Frequent anomaly correlation graph mining for root cause diagnose in micro-service architecture. In *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*, pages 1–8. IEEE, 2018.

[40] Meng Ma, Weilan Lin, Disheng Pan, and Ping Wang. Ms-rank: Multi-metric and self-adaptive root cause diagnosis for microservice applications. In *2019 IEEE International Conference on Web Services (ICWS)*, pages 60–67. IEEE, 2019.

[41] Meng Ma, Weilan Lin, Disheng Pan, and Ping Wang. Self-adaptive root cause diagnosis for large-scale microservice architecture. *IEEE Transactions on Services Computing*, 15(3):1399–1410, 2020.

[42] Meng Ma, Jingmin Xu, Yuan Wang, Pengfei Chen, Zonghua Zhang, and Ping Wang. Automap: Diagnose your microservice-based web applications automatically. In *Proceedings of The Web Conference 2020*, pages 246–258, 2020.

[43] Ping Wang, Jingmin Xu, Meng Ma, Weilan Lin, Disheng Pan, Yuan Wang, and Pengfei Chen. Cloudranger: Root cause identification for cloud native systems. In *2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pages 492–502. IEEE, 2018.

[44] Pooja Aggarwal, Ajay Gupta, Prateeti Mohapatra, Seema Nagar, Atri Mandal, Qing Wang, and Amit Paradkar. Localization of operational faults in cloud applications by mining causal dependencies in logs using golden signals. In *Service-Oriented Computing–ICSOC 2020 Workshops: AIOps, CFTIC, STRAPS, AI-PA, AI-IOTS, and Satellite Events, Dubai, United Arab Emirates, December 14–17, 2020, Proceedings*, pages 137–149. Springer, 2021.

[45] Álvaro Brandón, Marc Solé, Alberto Huélamo, David Solans, María S Pérez, and Victor Muntés-Mulero. Graph-based root cause analysis for service-oriented and microservice architectures. *Journal of Systems and Software*, 159:110432, 2020.

[46] Myunghwan Kim, Roshan Sumbaly, and Sam Shah. Root cause detection in a service-oriented architecture. *ACM SIGMETRICS Performance Evaluation Review*, 41(1):93–104, 2013.

[47] Areeg Samir and Claus Pahl. Dla: Detecting and localizing anomalies in containerized microservice architectures using markov models. In *2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 205–213. IEEE, 2019.

[48] Jörg Thalheim, Antonio Rodrigues, Istemi Ekin Akkus, Pramod Bhatotia, Ruichuan Chen, Bimal Viswanath, Lei Jiao, and Christof Fetzer. Sieve: Actionable insights from monitored metrics in distributed systems. In *Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference*, pages 14–27, 2017.

[49] Li Wu, Jasmin Bogatinovski, Sasho Nedelkoski, Johan Tordsson, and Odej Kao. Performance diagnosis in cloud microservices using deep learning. In *Service-Oriented Computing–ICSOC 2020 Workshops: AIOps, CFTIC, STRAPS, AI-PA, AI-IOTS, and Satellite Events, Dubai, United Arab Emirates, December 14–17, 2020, Proceedings*, pages 85–96. Springer, 2021.

[50] Vedika Agarwal, Rakshith Shetty, and Mario Fritz. Towards causal vqa: Revealing and reducing spurious correlations by invariant and covariant semantic editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, page 9687–9695, 2020.

[51] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):56–67, 2020.

[52] Hui Zhao, Jing An, Mengjie Yu, Diankai Lv, Kaida Kuang, and Tianqi Zhang. Nesterov-accelerated adaptive momentum estimation-based wavefront distortion correction algorithm. *Applied Optics*, 60(24):7177–7185, 2021.

[53] Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

[54] Robert Osazuwa Ness, Karen Sachs, Parag Mallick, and Olga Vitek. A bayesian active learning experimental design for inferring signaling networks. In *Research in Computational Molecular Biology: 21st Annual International Conference, RECOMB 2017, Hong Kong, China, May 3-7, 2017, Proceedings 21*, pages 134–156. Springer, 2017.

[55] Tianjian Zhang, Qian Chen, Yi Jiang, Dandan Miao, Feng Yin, Tao Quan, Qingjiang Shi, and Zhi-Quan Luo. Icassp-spgc 2022: Root cause analysis for wireless network fault localization. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 9301–9305. IEEE, 2022.

[56] Matej Zecevic, Devendra Singh Dhami, Petar Velickovic, and Kristian Kersting. Relating graph neural networks to structural causal models. *arXiv preprint arXiv:2109.04173*, 2021.

Table 3: Notations and their meanings.

Notation	SIZE	Meaning
$\mathcal{G}$		causal graph
$\boldsymbol{A}$	$m \times m$	the asymmetric adjacency matrix of observational DAG
$\mathbf{x}$	$m \times 1$	endogenous variables
$\mathbf{u}$	$m \times 1$	exogenous variables
$\mathrm{An}(i)$		the ancestor nodes of node i
$\mathbf{u}_{\mathrm{An}^*(i)}$		exogenous variables of the ancestor nodes of node i
$\mathrm{do}(\mathrm{x}_i = x_i)$		do-operation
$\mathcal{I}$		set of the intervened nodes
$\boldsymbol{A}_{\mathcal{I}}$	$m \times m$	the adjacency matrix of interventional DAG
$\mathbf{1}$	$m \times 1$	a column vector of all ones
$\odot$		Hadamard product
$q(\cdot)$		variational q-distribution
$p(\cdot)$		variational p-distribution
$\mathcal{N}$		normal distribution
Bern		Bernoulli distribution
$\mathrm{Lognormal}$		log-normal distribution
u	$m \times 1$	latent variables for normal distribution
$\boldsymbol{\gamma}$	$m \times 1$	latent variables for Bernoulli distribution
ϵ	1×1	global observation noise
ζ	1×1	inverse variance of the noise
μ_i	1×1	mean of normal distribution of node i
σ_i^2	1×1	variance of normal distribution of node i
π	1×1	probaility of taking one in a Bernoulli distribution
μ_ζ	$m \times 1$	mean of log-normal distribution
σ_ζ^2	$m \times 1$	variance of log-normal distribution
$\mu_i(\mathbf{x}, \boldsymbol{A})$	1×1	mean of the estimated normal distribution of node i
$\sigma_i^2(\mathbf{x}, \boldsymbol{A})$	1×1	variance of the estimated Normal distribution of node i
$\omega_i(\mathbf{x}, \boldsymbol{A})$	1×1	estimated probaility of taking one in a Bernoulli distribution of node i
$\mu_\zeta(\mathbf{x}, \boldsymbol{A})$	1×1	mean of the estimated log-normal distribution
$\sigma_\zeta^2(\mathbf{x}, \boldsymbol{A})$	1×1	variance of the estimated log-normal distribution
$p(\mathbf{x}, \mathbf{u}, \boldsymbol{\gamma}, \zeta \| \boldsymbol{A})$		the proposed gernerative model
$q(\mathbf{u}, \boldsymbol{\gamma}, \zeta \| \mathbf{x}, \boldsymbol{A})$		the proposed inference model
$\mathbb{E}_q$		expectation over the q-distribution
$\mathbb{H}_q$		entropy of the q-distribution
$\mathcal{L}$		the evidence lower bound (ELBO)
$D_{\mathrm{KL}}(q \|\| p)$		KL divergence between distributions q and p
$\mathrm{Dec}(\mathbf{u}, \boldsymbol{\gamma}, \boldsymbol{A})$		the proposed decoder

A Notations

See Table 3.

B More on Related Works

Causal Explanations: We also briefly review causal explanations as they are related to ITE. AI explainability endeavors to demystify model decisions, a pursuit encompassing feature attribution and contrastive explanations. Feature attribution methods initially focused on associations, revealing how input features correlate with predictions [24]. Moving beyond mere correlations, recent approaches integrate causality to enhance interpretability: CXPlain [32] employs supervised learning to discern the causal impact of features on model predictions, though computational demands escalate with the need for repeated model evaluations. Generative Causal Explanations (GCE) [33] introduce disentangled latent factors to isolate causal effects, yet ensuring these factors accurately reflect the data distribution is complex. Asymmetric Shapley Value (ASV) [25], Shapley Flows [26], and PWSHAP [27] further this trend by considering the causal graphs, with the former adapting Shapley values to reflect causal structures and the latter two focusing on the causal relationships signified by the graph's edges and paths. On the other hand, Contrastive explanations, illustrated by Counterfactual Explanations (CE) [34] and Causal Algorithmic Recourse (CAR) [35, 36], offer a counterfactual narrative on how slight modifications or specific interventions could lead to different, often more favorable outcomes. Specifically, CE suggests minimal feature changes for an alternate outcome, teaching individuals how to achieve a different result. Their application must navigate constraints of plausibility and actionability to avoid recommending impractical changes. CAR extends CEs by proposing interventions grounded in causal relationships, aiming to recalibrate outcomes with consideration of the cost and effect of actions.

These causal explanation methods intersect with ITE. They can serve as a foundation for ITE by treating the most influential factors as potential intervention targets. Conversely, ITE can reciprocate by informing causal explanations since it pinpoints the very intervention targets that are the roots of observed outcomes.

Root Cause Analysis: Lastly, graph-based RCA methods warrant discussion, due to the intertwined nature of root causes and intervention targets. These methods usually operate in two distinct stages. Initially, the graph structure is established either via causal discovery algorithms such as the PC algorithm [28, 37, 38, 39, 40, 41, 42, 43], Granger causality [44] and DAG-GNN [31], or it is derived from domain-specific knowledge like topology graphs [45, 46, 29, 47, 48, 49, 30]. The second stage then leverages algorithms such as PageRank [46, 31], breadth-first or depth-first search [28, 37, 38, 29], and random walk [50, 40, 41, 42, 43, 30] for root cause localization within the graph. Despite their utility, the second stage tends to focus on association rather than causation, assigning a higher correlation to the connections between a node and its parents over those between the node and its children to consider the directionality in the causal graph. In contrast, as highlighted in the preceding sections, CI-RCA [4] emphasizes causation, utilizing linear regression-based hypothesis testing to pinpoint intervention targets. As an alternative, RCD [9] incorporates the Ψ-PC algorithm in a hierarchical fashion, intertwining the learning of intervention targets with graph structure discovery. These approaches not only improve root cause identification but also harmonize with ITE goals, fostering a causally informed analysis in RCA. However, as noted earlier, these methods lack collaborative learning and are fully unsupervised, necessitating complete inference for each RCA instance from scratch and failing to utilize labeling information.

C Proof of Proposition 1

According to the Neumann power series for the matrix inverse, we can obtain:

$$(\boldsymbol{I} - \boldsymbol{A}^T)^{-1} = \sum_{k=0}^{\infty} (\boldsymbol{A}^k)^T, \tag{15}$$

where A^k denotes the k-th power of the matrix A, which involves multiplying the matrix A by itself k times. In the above expression, entry (i, j) in the k-th power of $\boldsymbol{A}$ can be elaborated as:

$$\begin{aligned}(\boldsymbol{A}^k)_{ij} &= \sum_{w_1,\ldots,w_{k-1}} \boldsymbol{A}_{i,w_1} \boldsymbol{A}_{w_1,w_2} \ldots \boldsymbol{A}_{w_{k-1},j} \\ &= \sum \mathrm{path}_k(i, j),\end{aligned} \tag{16}$$

where the sum encapsulates all paths $(i, w_1, w_2, \ldots, w_{k-1}, j)$ from i to j with length k. In directed graphs, these paths must observe edge directionality. Consequently, for DAGs, the matrix $\boldsymbol{A}^k$ becomes a zero matrix when k surpasses the graph diameter δ, as no paths of length k exist between any two nodes within such graphs. Substitute (15) into (5), and the GNN layer is recast as:

$$\mathbf{x} = f_2\Big(\sum_{k=0}^{\infty} (\boldsymbol{A}^k)^T f_1(\mathbf{u})\Big). \tag{17}$$

By multiplying $(\boldsymbol{A}^k)^T$ with $f_1(\mathbf{u})$ as in (17), node i can receive information from its ancestors that have a path of length k connecting to i. As k goes from 0 to ∞, node i accrues information from the input features of its own and all its ancestors. On the other hand, paths only exist between node i and one of its ancestors. Conversely, such paths are non-existent between node i and non-ancestral nodes; hence, node i exclusively assimilates inputs from its ancestors, thereby concluding the proof.

D Proof of Proposition 2

In consideration of Proposition 1 and given the decoder in Eq. (5), it is evident that x_i is a function solely of $\mathbf{u}_{\mathrm{An}^*(i)}$ for all indices i. Consequently, the probability distribution $p(\mathrm{x}|\mathrm{u}, \boldsymbol{A})$ can be factorized into:

$$p(\mathrm{x}|\mathrm{u}, \boldsymbol{A}) = \prod_i p(\mathrm{x}_i|\mathbf{u}_{\mathrm{An}^*(i)}), \tag{18}$$

which logically concludes the proof.

E Proof of Proposition 3

The essence of a causal intervention lies in severing all the incoming edges to the intervened nodes. Therefore, the decoder stipulated in Eq. (5) can faithfully represent causal interventions only if it encompasses all possible causal pathways; otherwise, severing certain pathways would exert no influence on the resulting intervention configuration. The decoder in Eq. (5) does indeed model all causally relevant paths, as corroborated by Proposition 1, thereby completing the proof.

F Derivation of the ELBO

Recall that the generative model (i.e., the p distribution) can be factorized as:

$$p(\mathbf{x}, \mathbf{u}, \boldsymbol{\gamma}, \zeta|\boldsymbol{A}) = p(\mathbf{x}|\mathbf{u}, \boldsymbol{\gamma}, \boldsymbol{A}, \zeta)p(\zeta)\prod_{i=1}^{m} p(\mathrm{u}_i)p(\gamma_i), \tag{19}$$

where

$$p(\mathrm{u}_i) = \mathcal{N}(0, 1), \quad \forall i, \tag{20}$$

$$p(\gamma_i) = \mathrm{Bern}(\pi), \quad \forall i, \tag{21}$$

$$p(\zeta) \propto 1/\zeta, \tag{22}$$

$$p(\mathbf{x}|\mathbf{u}, \boldsymbol{\gamma}, \boldsymbol{A}, \zeta) = \mathcal{N}\big(\mathrm{Dec}(\mathbf{u}, \boldsymbol{\gamma}, \boldsymbol{A}), \zeta^{-1}\boldsymbol{I}\big). \tag{23}$$

, and the inference model (i.e., the q distribution) can be factorized as:

$$q(\mathbf{u}, \boldsymbol{\gamma}, \zeta|\mathbf{x}, \boldsymbol{A}) = q(\zeta|\mathbf{x}, \boldsymbol{A})\prod_{i=1}^{m} q(\mathrm{u}_i|\mathbf{x}, \boldsymbol{A})q(\gamma_i|\mathbf{x}, \boldsymbol{A}), \tag{24}$$

where

$$q(\mathrm{u}_i|\mathbf{x}, \boldsymbol{A}) = \mathcal{N}\big(\mu_i(\mathbf{x}, \boldsymbol{A}), \sigma_i^2(\mathbf{x}, \boldsymbol{A})\big), \tag{25}$$

$$q(\gamma_i|\mathbf{x}, \boldsymbol{A}) = \mathrm{Bern}\,\big(\omega_i(\mathbf{x}, \boldsymbol{A})\big), \tag{26}$$

$$q(\zeta|\mathbf{x}, \boldsymbol{A}) = \mathrm{Lognormal}\,\big(\mu_\zeta(\mathbf{x}, \boldsymbol{A}), \sigma_\zeta^2(\mathbf{x}, \boldsymbol{A})\big). \tag{27}$$

Note that the parameters of the above q distributions are explicit functions of the given sample of the endogenous variables x and the adjacency matrix $\boldsymbol{A}$, which is parameterized by the GAT-based inference network. By substituting the p (19) and q distributions (24) into the ELBO (14), we can obtain:

$$\begin{aligned}
\mathcal{L} =& \mathbb{E}_q\Big[\log p(\mathbf{x}|\mathbf{u}, \boldsymbol{\gamma}, \boldsymbol{A}, \zeta) + \log p(\zeta) + \sum_{i=1}^{m}\big(\log p(\mathrm{u}_i) + \log p(\gamma_i)\big)\Big] + \mathbb{H}_q, \\
=& \mathbb{E}_q\Big[\log p(\mathbf{x}|\mathbf{u}, \boldsymbol{\gamma}, \boldsymbol{A}, \zeta) + \log p(\zeta) - \log q(\zeta|\mathbf{x}, \boldsymbol{A}) + \\
& \sum_{i=1}^{m}\big(\log p(\mathrm{u}_i) - \log q(\mathrm{u}_i|\mathbf{x}, \boldsymbol{A}) + \log p(\gamma_i) - \log q(\gamma_i|\mathbf{x}, \boldsymbol{A})\big)\Big], \\
=& \mathbb{E}_q\big[\log p(\mathbf{x}|\mathbf{u}, \boldsymbol{\gamma}, \boldsymbol{A}, \zeta)\big] - D_{\mathrm{KL}}\big(q(\zeta|\mathbf{x}, \boldsymbol{A})\|p(\zeta)\big) -
\end{aligned}$$

$$\sum_{i=1}^{m}\Big(D_{\mathrm{KL}}\big(q(\mathrm{u}_i|\mathbf{x},\boldsymbol{A})\|p(\mathrm{u}_i)\big)+D_{\mathrm{KL}}\big(q(\gamma_i|\mathbf{x},\boldsymbol{A})\|p(\gamma_i)\big)\Big), \tag{28}$$

where D_{KL} denotes the KL (Kullback-Leibler) divergence between two distributions. We now delve into the expectation term in the above expression, which is given by:

$$\begin{aligned}
&\mathbb{E}_q\big[\log p(\mathbf{x}|\mathbf{u},\boldsymbol{\gamma},\boldsymbol{A},\zeta)\big]\\
=&\frac{m}{2}\langle\log\zeta\rangle-\frac{1}{2}\langle\zeta\rangle\langle(\mathbf{x}-\mathrm{Dec}(\mathbf{u},\boldsymbol{\gamma},\boldsymbol{A}))^T(\mathbf{x}-\mathrm{Dec}(\mathbf{u},\boldsymbol{\gamma},\boldsymbol{A}))\rangle,\\
=&\frac{m}{2}\mu_\zeta(\mathbf{x},\boldsymbol{A})-\frac{1}{2}\exp\Big(\mu_\zeta(\mathbf{x},\boldsymbol{A})+\frac{\sigma_\zeta^2(\mathbf{x},\boldsymbol{A})}{2}\Big)\cdot\\
&\langle(\mathbf{x}-\mathrm{Dec}(\mathbf{u},\boldsymbol{\gamma},\boldsymbol{A}))^T(\mathbf{x}-\mathrm{Dec}(\mathbf{u},\boldsymbol{\gamma},\boldsymbol{A}))\rangle.
\end{aligned} \tag{29}$$

The remaining three KL divergence terms can be written as:

$$D_{\mathrm{KL}}\big(q(\zeta|\mathbf{x},\boldsymbol{A})\|p(\zeta)\big)=-\frac{1}{2}\log\sigma_\zeta^2(\mathbf{x},\boldsymbol{A}), \tag{30}$$

$$D_{\mathrm{KL}}\big(q(\mathrm{u}_i|\mathbf{x},\boldsymbol{A})\|p(\mathrm{u}_i)\big)=\frac{1}{2}\big(\mu_i^2(\mathbf{x},\boldsymbol{A})+\sigma_i^2(\mathbf{x},\boldsymbol{A})-\log\sigma_i^2(\mathbf{x},\boldsymbol{A})\big), \tag{31}$$

$$\begin{aligned}D_{\mathrm{KL}}\big(q(\gamma_i|\mathbf{x},\boldsymbol{A})\|p(\gamma_i)\big)=&\,\omega_i(\mathbf{x},\boldsymbol{A})\,\mathrm{logit}(\omega_i(\mathbf{x},\boldsymbol{A}))-\omega_i(\mathbf{x},\boldsymbol{A})\,\mathrm{logit}(\pi)\\&+\log(1-\omega_i(\mathbf{x},\boldsymbol{A}))-\log(1-\pi).\end{aligned} \tag{32}$$

Algorithm 1 DeepITE Semi-supervised Training Algorithm

```
Require: Causal graph $\mathcal{G}$, adjacency matrix $\boldsymbol{A}$, endogenous variables $\mathbf{x}^{\{1:m\}}$, labels of intervention targets $\boldsymbol{\gamma}_{\text{true}}^{\{1:m\}}$ if available.
Ensure: Parameters of the inference network $\phi$, parameters of the generative network $\theta$;
Initialize $\phi, \theta$ randomly;
repeat
    Pass $\mathbf{x}^{\{1:m\}}$ and $\boldsymbol{A}$ through the inference network to get the parameters for $q(\mathbf{u}|\mathbf{x}^{\{1:m\}}, \boldsymbol{A})$, $q(\boldsymbol{\gamma}|\mathbf{x}^{\{1:m\}}, \boldsymbol{A})$, $q(\zeta|\mathbf{x}^{\{1:m\}}, \boldsymbol{A})$;
    Draw samples from the Normal distribution $q(\mathbf{u}|\mathbf{x}^{\{1:m\}}, \boldsymbol{A})$;
    Draw samples from the Bernoulli distribution $q(\boldsymbol{\gamma}|\mathbf{x}^{\{1:m\}}, \boldsymbol{A})$ using the gumbel-softmax reparameterization trick;
    Draw samples from the Log Normal distribution $q(\zeta|\mathbf{x}^{\{1:m\}}, \boldsymbol{A})$;
    $\widehat{\mathbf{x}}^{\{1:m\}} \leftarrow \mathrm{Dec}(\boldsymbol{u}^{\{1:m\}}, \boldsymbol{\gamma}^{\{1:m\}}, \zeta, \boldsymbol{A})$
    if $\boldsymbol{\gamma}_{\text{true}}^{\{1:m\}}$ is available then
        Compute the negative ELBO (28) and the maximum log likelihood of $\boldsymbol{\gamma}_{\text{true}}^{\{1:m\}}$;
    else
        Compute the negative ELBO (28);
    end if
    Update $\phi, \theta$ via gradient descent;
until convergence
return $\phi, \theta$
```

G Experiment Details

G.1 Experiment Setup

Unless otherwise specified, in all of our experiments for DeepITE, we set the hidden dimension in GAT and MLP to 64. For optimization, we used NAdam [52] with a learning rate 1×10^{-4}. We conducted training for 1000 epochs and select the checkpoints with the lowest training loss. The temperature $\mathbf{t}$ for gumbel-softmax [23] is calculated by $\mathbf{t} = 101 - 0.2\mathbf{e}$ when epoch $\mathbf{e} <= 500$ and $\mathbf{t} = 0.5/(e - 500)$ for epoch $\mathbf{e} > 500$. All the training runs on 4 NVIDIA TESLA P100 GPUs with 50GB of VRAM. All the inference runs on a MacBook Pro 16 inch with a 6-core Intel i7 CPU and 16 GB of RAM.

The key assumptions and characteristics of the comparative methods, as well as the time complexity in the inference process, are summarized in the table 4 and table 5 To facilitate a fair comparison, all methods (including ITE, RCA, and XAI) are provided with the same ground truth causal graph,

Table 4: Key assumptions and characteristics of comparing methods. Here, m and n denote the number of nodes and edges in the DAG, $p_\triangle$ is the number of intervention targets given by the precision different estimation algorithm, and finally, T, L, and D represent the number of trees, the depth of the trees, and the number of leaf nodes in the gradient-boosted trees.

METHOD	CAUSAL GRAPH	REFERENCE SET	INTERVENTION	CONFOUNDER	AMORTIZATION	GRAPH SIZE	TIME COMPLEXITY
DeepITE	Given	No	Soft&Hard	No	Yes	< 1000	$\mathcal{O}(m+n)$
UT-IGSP	Unknown	Require	Soft	No	No	< 100	$\mathcal{O}(2^{m-1})$
CITE	Given	Require	Soft	No	No	< 100	$\mathcal{O}\left(2^{p_\triangle}\right)$
PreDITEr	Given	Require	Soft	No	Yes	< 100	$\mathcal{O}\left(2^{p_\triangle}\right)$
LIT	Unknown	Require	Soft	No	Yes	< 100	$\mathcal{O}(m^2)$
CauseInfer	Given	No	Hard	No	Yes	< 1000	$\mathcal{O}(m^2)$
MicroHECL	Given	No	Hard	No	Yes	< 1000	$\mathcal{O}(m^2)$
MicroRCA	Given	No	Hard	No	Yes	< 1000	$\mathcal{O}(m^2)+O(m+n)$
CausalRCA	Given	No	Hard	No	Yes	< 1000	$\mathcal{O}(m^2)$
CI-RCA	Given	No	Hard	No	Yes	< 1000	$\mathcal{O}(m^2)$
RCD	Given	No	Hard	No	Yes	< 1000	$\mathcal{O}(m^3)$
TreeExplainer	Unknown	No	Soft	No	No	< 100	$\mathcal{O}\left(TLD^2\right)$
ASV	Given	No	Soft	No	No	< 100	$\mathcal{O}(m^3)$
ShapleyFlow	Given	No	Soft	No	No	< 100	$\mathcal{O}(m^3)$
PWSHAP	Given	No	Soft	Yes	No	< 100	$\mathcal{O}(m^3)$

Table 5: Time Complexity during Inference. Here, m and n denote the number of nodes and edges in the DAG, $p_\triangle$ is the number of intervention targets given by the precision different estimation algorithm, and finally, T, L, and D represent the number of trees, the depth of the trees, and the number of leaf nodes in the gradient-boosted trees.

METHOD	TIME COMPLEXITY
UT-IGSP [10]	$\mathcal{O}(2^{m-1})$
CITE [3]	$\mathcal{O}\left(2^{p_\triangle}\right)$
PreDITEr [6]	$\mathcal{O}\left(2^{p_\triangle}\right)$
CauseInfer [28]	$\mathcal{O}(m^2)$
MicroHECL [29]	$\mathcal{O}(m^2)$
MicroRCA [30]	$\mathcal{O}(m^2)+O(m+n)$
CausalRCA [31]	$\mathcal{O}(m^2)$
CI-RCA [4]	$\mathcal{O}(m^2)$
RCD [9]	$\mathcal{O}(m^3)$
TreeExplainer [51]	$\mathcal{O}\left(TLD^2\right)$
ASV [25]	$\mathcal{O}(m^3)$
ShapleyFlow [26]	$\mathcal{O}(m^3)$
PWSHAP [27]	$\mathcal{O}(m^3)$
DeepITE	$\mathcal{O}(m+n)$

eschewing the need for graph construction from data for some RCA methods. Within the realm of XAI, constructing a forward predictive model is a prerequisite for backward attribution analysis—the process used to identify the intervention targets. To facilitate this, we construct a predictive model using gradient-boosted trees. This forward model is trained to predict whether the graph is intervened based on the full set of node features $\mathbf{x}$, in a supervised manner. During inference, we extract the intervention targets by identifying the top k nodes that yield the highest attribution scores.

G.2 Synthetic Data Generation

The generation process begins by creating a random adjacency matrix $\boldsymbol{A}$, which is structured to be upper-triangular to ensure the resulting DAG is indeed acyclic. The matrix's non-zero entries are uniformly distributed across the range $[-2,-0.5]\cup[0.5,2]$, representing the possible strengths of causal relationships between nodes. For each node within the DAG, time series data are synthesized according to a model that captures causal dependencies, as inspired by the linear decoder equation: $\mathbf{x}(t)=(\boldsymbol{I}-\boldsymbol{A}^T)^{-1}(\mathbf{u}(t)+\beta\mathbf{x}(t-1))$, where t indicates the discrete time steps, and β denotes the autoregressive coefficient, influencing the temporal consistency of the data.

Interventions are then introduced in the time series at time t, with the set of intervention nodes $\mathcal{I}$ being randomly selected from all non-root nodes, and the size of $\mathcal{I}$ adhering to a Poisson distribution. For nodes within $\mathcal{I}$, we augment the corresponding exogenous noise, adhering to the three-sigma rule for significant deviation. Each generated time series spans 1000 time steps, with interventions

Table 6: Recall@k of different algorithms for detecting the intervened nodes from the Synthetic dataset following the classical ITE setting.

DATASET	**Linear-10**		**Linear-20**		**Nonlinear-10**		**Nonlinear-20**	
METRICS	Recall@1	Recall@3	Recall@1	Recall@3	Recall@1	Recall@3	Recall@1	Recall@3
UT-IGSP	0.921 ± 0.009	0.972 ± 0.005	0.917 ± 0.007	0.956 ± 0.005	0.933 ± 0.008	0.989 ± 0.004	0.927 ± 0.005	0.972 ± 0.004
CITE	0.939 ± 0.007	0.989 ± 0.003	0.933 ± 0.009	0.961 ± 0.004	0.789 ± 0.011	0.956 ± 0.006	0.733 ± 0.010	0.883 ± 0.005
PreDITEr	0.944 ± 0.006	0.989 ± 0.003	0.933 ± 0.008	0.967 ± 0.006	0.694 ± 0.005	0.861 ± 0.004	0.561 ± 0.004	0.822 ± 0.003
TreeExplainer	0.764 ± 0.015	0.897 ± 0.020	0.551 ± 0.010	0.809 ± 0.018	0.765 ± 0.015	0.909 ± 0.021	0.539 ± 0.010	0.773 ± 0.017
ASV	0.751 ± 0.014	0.821 ± 0.019	0.629 ± 0.011	0.782 ± 0.017	0.679 ± 0.014	0.737 ± 0.018	0.510 ± 0.010	0.554 ± 0.016
ShapleyFlow	0.858 ± 0.017	0.928 ± 0.022	0.817 ± 0.012	0.858 ± 0.019	0.841 ± 0.016	0.919 ± 0.023	0.811 ± 0.011	0.870 ± 0.020
PWSHAP	0.784 ± 0.016	0.892 ± 0.021	0.650 ± 0.011	0.815 ± 0.018	0.776 ± 0.014	0.895 ± 0.015	0.647 ± 0.010	0.781 ± 0.014
CauseInfer	0.869 ± 0.001	0.935 ± 0.001	0.430 ± 0.003	0.615 ± 0.003	0.720 ± 0.002	0.819 ± 0.001	0.561 ± 0.004	0.692 ± 0.003
MicroHECL	0.407 ± 0.011	0.624 ± 0.012	0.387 ± 0.014	0.485 ± 0.019	0.420 ± 0.012	0.660 ± 0.015	0.411 ± 0.017	0.536 ± 0.022
MicroRCA	0.759 ± 0.002	0.890 ± 0.001	0.730 ± 0.003	0.846 ± 0.002	0.398 ± 0.002	0.527 ± 0.002	0.175 ± 0.002	0.233 ± 0.002
CausalRCA	0.820 ± 0.003	0.885 ± 0.002	0.705 ± 0.002	0.841 ± 0.001	0.729 ± 0.003	0.805 ± 0.003	0.545 ± 0.002	0.608 ± 0.001
CI-RCA	0.865 ± 0.002	0.941 ± 0.001	0.734 ± 0.003	0.923 ± 0.003	0.821 ± 0.003	0.896 ± 0.002	0.616 ± 0.002	0.683 ± 0.002
RCD	0.803 ± 0.004	0.871 ± 0.005	0.695 ± 0.005	0.837 ± 0.007	0.713 ± 0.005	0.801 ± 0.006	0.540 ± 0.006	0.597 ± 0.006
DeepITE	0.999 ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000	0.998 ± 0.001	**1.000** ± 0.000	**0.999** ± 0.000	**0.999** ± 0.000

occurring randomly between time steps 100 and 900. Data preceding the intervention time t are labeled as observational, while data from t onwards are categorized as interventional. We explore causal graphs of varying complexity, with 50, 100, and 500 nodes, and associated edge counts of 100, 500, and 5000, respectively. To ensure a comprehensive assessment, 10 unique graphs are generated for each size, and for each graph, we create 100 distinct instances with varying intervention targets. The dataset is partitioned with an 85:5:10 ratio for training, validation, and testing.

G.3 Synthetic Data under the Classical ITE Setting

For an equitable evaluation with existing ITE approaches, we further devise experiments on synthetically generated datasets tailored to classical ITE configurations, using dataset generation scripts from VACA [21]. Our setup included two linear and two non-linear SCMs. Initially, we generate 10,000 observational samples from each SCM. Subsequently, we perform ten distinct interventions on randomly selected variables, mimicking the procedures followed by ITE methods. Each intervention resulted in the creation of 1,000 samples. Throughout these interventions, we maintain the SCMs' modularity and provid the causal graph with the data. We divide the dataset into training, validation, and testing batches with an 85:5:10 ratio, respectively.

Table 6 reveals that DeepITE consistently surpasses competing methods in performance. Notably, traditional ITE approaches like UT-IGSP, CITE, and PreDITEr benefit significantly from an adequate pool of observational samples on small-scale graphs, reflecting a marked improvement in their effectiveness. Similarly, XAI techniques—ShapleyFlow in particular—demonstrate commendable performance, proving to be well-adapted for smaller graphs where their capabilities can be fully leveraged. Conversely, RCA methods exhibit weaker results, a trend that may be linked to their design inclination towards larger-scale graphs. These larger configurations are often seen in complex environments such as microservice diagnosis, suggesting that RCA methods' strengths are not fully utilized in smaller or less complex scenarios highlighted in this comparison.

Furthermore, it is evident that models based on the premise of linear SCMs, such as CITE, PreDITEr, and CI-RCA, tend to underperform in settings that demand an understanding of non-linear dynamics. In contrast, DeepITE, which utilizes learnable SCMs, demonstrates impressive versatility by effectively addressing both linear and non-linear scenarios.

G.4 Real Data Description

Protein Signaling Dataset: The well-known protein signaling dataset, which originates from Sachs *et al.* [53], investigates the complex interactions within T-4 cell signaling networks. The dataset comprises 11 nodes and 16 edges, with a collection of 1755 observational and 4091 interventional samples derived from five different experimental environments where various drugs were used to modulate signaling proteins. We harness an accepted ground truth network structure in [54], and the preprocessing steps outlined in [10], to benchmark DeepITE's performance with other models. The dataset is partitioned with an 85:5:10 ratio for training, validation, and testing.

Table 7: Impact of Labeled Data on DeepITE (mix) for the Synthetic Dataset. The proportion of labeled data is shown in the brackets.

DATASET	**HeterITE-50**		**HeterITE-100**		**HeterITE-500**	
METRICS	Recall@1	Recall@5	Recall@1	Recall@5	Recall@1	Recall@5
DeepITE (0%)	0.718 ± 0.004	0.945 ± 0.001	0.690 ± 0.004	0.923 ± 0.002	0.627 ± 0.004	0.875 ± 0.002
DeepITE (5%)	0.820 ± 0.002	0.986 ± 0.002	0.731 ± 0.003	0.960 ± 0.001	0.667 ± 0.003	0.922 ± 0.002
DeepITE (10%)	0.821 ± 0.002	0.991 ± 0.001	0.728 ± 0.003	0.944 ± 0.003	0.674 ± 0.003	0.925 ± 0.002
DeepITE (25%)	0.829 ± 0.002	0.995 ± 0.001	0.737 ± 0.002	0.982 ± 0.001	0.672 ± 0.004	0.954 ± 0.001
DeepITE (50%)	0.856 ± 0.002	0.998 ± 0.000	0.750 ± 0.003	0.996 ± 0.001	0.678 ± 0.003	0.952 ± 0.002
DeepITE (75%)	**0.873** ± 0.001	**1.000** ± 0.000	**0.762** ± 0.003	**0.998** ± 0.000	**0.699** ± 0.004	**0.966** ± 0.002
DeepITE (100%)	0.869 ± 0.002	0.999 ± 0.001	0.773 ± 0.002	**0.998** ± 0.001	0.697 ± 0.002	**0.966** ± 0.001

Table 8: Impact of Labeled Data on DeepITE for the two Real Datasets. The proportion of labeled data is shown in the brackets.

DATASET	**Protein Signaling**			**ICASSP-SPGC 2022**	
METRICS	Recall@1	Recall@1	Recall@5	Root.Acc	Score
DeepITE (0%)	0.652 ± 0.004	0.881 ± 0.002	0.984 ± 0.001	0.9794 ± 0.0054	0.9085 ± 0.0101
DeepITE (5%)	0.842 ± 0.002	0.906 ± 0.001	0.994 ± 0.001	**0.9964** ± 0.0023	**0.9524** ± 0.0040
DeepITE (10%)	0.850 ± 0.003	0.920 ± 0.001	0.994 ± 0.001	**0.9964** ± 0.0000	**0.9524** ± 0.0000
DeepITE (25%)	0.849 ± 0.002	0.922 ± 0.001	0.994 ± 0.000	**0.9964** ± 0.0000	**0.9524** ± 0.0000
DeepITE (50%)	0.868 ± 0.002	**0.925** ± 0.000	**0.995** ± 0.000	**0.9964** ± 0.0000	**0.9524** ± 0.0000
DeepITE (75%)	0.863 ± 0.002	0.924 ± 0.001	**0.995** ± 0.000	**0.9964** ± 0.0000	**0.9524** ± 0.0000
DeepITE (100%)	**0.872** ± 0.001	**0.925** ± 0.000	**0.995** ± 0.000	**0.9964** ± 0.0000	**0.9524** ± 0.0000

ICASSP-SPGC 2022[5]: The ICASSP-SPGC 2022 dataset [55], derived from active 5G networks, is a real-world telecommunications dataset for RCA comprising 2984 samples and 23 variables that represent various Key Performance Indicators (KPIs). Human experts have verified the accompanying causal graph, yet only around 45% of the data are explicitly labeled with root cause faults. Unlike the previously mentioned dataset, where interventions are directly known from labels, the root causes here are represented by unobserved variables outside the causal graph. However, experts provide a mapping of each root cause to observable causal variables. During training with labeled data, we treat these associated variables as if they had been intervened upon. Additionally, 600 extra samples are provided for testing purposes.

G.5 Semi-Supervised Learning

In this section, we delve into the influence of labeled data on the efficacy of DeepITE, with our findings summarized in Tables 7-8. Across all datasets under consideration, it is clear that the incorporation of labeled data yields a substantial enhancement in DeepITE's performance, with improvements in Recall@1 ranging between 4% to 20%, depending on the dataset. Notably, the most pronounced gains are observed when the initial 5% to 10% of labeled data are integrated, with the rate of improvement tapering off beyond this point. This suggests that even a modest quantity of labeled data can lead to significant performance boosts, a fact that bears particular relevance in practical scenarios where acquiring a limited amount of labeled data is typically feasible and can offer considerable benefits to DeepITE. Conversely, the other baseline methods in our study do not possess the capability to leverage such labeling information to their advantage.

G.6 Ablation Study

The ablation study of DeepITE, in comparison to VACA and DAG-GNN, is presented in Table 9. In Section 5.2, we have delved into the relationships between DeepITE, DAG-GNN, and VACA. We highlighted the limitations of DAG-GNN's inference model and VACA's generative model and illustrated how DeepITE overcomes these shortcomings. DAG-GNN utilizes an inference model represented as $\mathbf{u} = f_4((\boldsymbol{I} - \boldsymbol{A}^T) f_3(\mathbf{x}))$, which has constraints as it can only gather messages from a node's parents. On the other hand, DeepITE's inference model is crafted to aggregate messages from all nodes within the Markov blanket of a given node, ensuring a more flexible inference model tailored

[5] https://www.aiops.sribd.cn/home/statement

Table 9: Ablation study

		MMD(Obs)	MMD(Int)	SSE
Graph-50	VACA	3.90 ± 0.17	59.3 ± 5.3	3742.46 ± 495.35
	DAG-GNN	4.47 ± 0.26	67.80 ± 7.14	4525.58 ± 539.04
	DeepITE (VACA Decoder)	1.93 ± 0.38	10.78 ± 2.99	2534.70 ± 105.22
	DeepITE (DAG-GNN Encoder)	2.58 ± 0.40	32.80 ± 7.89	3177.97 ± 189.61
	DeepITE	**0.12** ± 0.075	**0.76** ± 0.051	**415.66** ± 26.92
Graph-100	VACA	4.58 ± 0.26	70.89 ± 4.14	4333.90 ± 584.71
	DAG-GNN	5.41 ± 0.30	89.71 ± 6.33	5410.52 ± 656.23
	DeepITE (VACA Decoder)	2.14 ± 0.15	14.7 ± 1.53	2967.47 ± 151.69
	DeepITE (DAG-GNN Encoder)	3.13 ± 0.29	73.92 ± 5.96	3651.14 ± 245.47
	DeepITE	**0.19** ± 0.073	**0.96** ± 0.066	**490.06** ± 37.15
Graph-500	VACA	5.36 ± 0.54	116.30 ± 2.59	6846.90 ± 795.12
	DAG-GNN	7.39 ± 0.46	154.96 ± 3.63	7511.26 ± 916.93
	DeepITE (VACA Decoder)	3.48 ± 0.16	31.89 ± 5.08	4391.59 ± 340.15
	DeepITE (DAG-GNN Encoder)	4.61 ± 0.22	48.60 ± 8.20	5316.33 ± 454.54
	DeepITE	**1.08** ± 0.045	**5.16** ± 0.65	**731.17** ± 75.32

for ITE tasks. VACA employs a generative model with a minimum requirement of $\delta - 1$ MPNN layers, where δ represents the graph diameter. This limitation restricts the propagation distance of information within the graph, hindering its performance in estimating distributions over large graphs. DeepITE distinguishes itself by employing the generative model, specified in Eq. (7), thereby overcomes the limitations imposed by VACA's dependence on graph diameter.

To demonstrate the superiority of DeepITE's generative and inference models, we devised two ablation designs by seperately modifying DeepITE's encoder layers to DAG-GNN's encoder and DeepITE's decoder layers to VACA's decoder. These settings were compared alongside VACA and DAG-GNN. Based on the synthetic data in Appendix G.2, we combined the observational and interventional data and fed them into the model along with the adjacency matrix $\boldsymbol{A}$ for observational data, which is exactly how ITE works. Since VACA and DAG-GNN do not directly output ITE results, we utilized Maximum Mean Discrepancy (MMD) and the standard deviation of the squared error (SSE) between the true and estimated values for our evalutaion, providing another dimension to gauge their effectiveness in estimating ITE. MMD was calculated separately for observational and interventional data, even though this information was unknown to the models.

The results, as shown in Table 9, revealed DeepITE outperforming the other models, validating the excellence of DeepITE's generative and inference models. Due to its relatively inflexible model design, DAG-GNN struggles to effectively reconstruct the biases associated with intervention points. The constraint of minimal number decoder layers limits VACA's capability in capturing long-range dependencies and interactions within the graph structure, leading to a significant decrease in its performance as the number of graph nodes increase. Although the ablation methods showed suboptimal performance, the flexibility introduced by the ITE indicator γ enabled them to outperform the origin VACA and DAG-GNN. This underscores the adaptability of our approach for ITE tasks despite its limitations in certain scenarios.

G.7 Case Study

To address the challenges of root cause analysis in complex systems with interconnected variables, we conducted a case study on the real-world dataset ICASSP-SPGC 2022, evaluating the performance of DeepITE in comparison to other methods. The groundtruth causal graph is provided in [55]. Recall that we focus on the RCA problem and we aim to identify and present the root causes of the system to users. Note that while we have labels for the root causes in our testing data, we only possess observations for the observable variables represented in the graph. Consequently, all methodologies employed can only localize observable variables as intervention targets (ITs), rather than directly identifying the root causes. For instance, RootCause 2 (a weak signal in marginal areas) can influence feature19, feature X, and feature Y. However, RootCause 3 also affects feature X. As a result, when

Table 10: Case study on three different samples (1758, 1760, and 1093) from the real-world dataset ICASSP-SPGC 2022. ITE Top-k represents the k identified features. The root causes are then inferred from these k features. The ground truths for these samples are rootcause2, rootcause3, and rootcause2&rootcause3.

	SAMPLE 1758		SAMPLE 1760		SAMPLE 1093	
	ITE Top-1	**ROOT CAUSE**	**ITE Top-1**	**ROOT CAUSE**	**ITE Top-2**	**ROOT CAUSE**
CauseInfer	FeatureY	Unspecified	FeatureY	Unspecified	Feature2&FeatureX	Unspecified
MicroHECL	FeatureX	Unspecified	FeatureY	Unspecified	FeatureX&Feature60	RootCause3
MicroRCA	FeatureY	Unspecified	FeatureY	Unspecified	FeatureX&Feature60	RootCause3
CausalRCA	FeatureY	Unspecified	FeatureX	Unspecified	FeatureY&FeatureX	Unspecified
CI-RCA	FeatureY	Unspecified	FeatureX	Unspecified	FeatureX&Feature60	RootCause3
RCD	FeatureY	Unspecified	FeatureX	Unspecified	FeatureY&FeatureX	Unspecified
TreeExplainer	FeatureX	Unspecified	Feature1	Unspecified	FeatureX&Feature1	Unspecified
ASV	FeatureY	Unspecified	FeatureX	Unspecified	FeatureY&FeatureX	Unspecified
ShapleyFlow	FeatureY	Unspecified	Feature17	Unspecified	FeatureY&FeatureX	Unspecified
PWSHAP	FeatureY	Unspecified	FeatureX	Unspecified	FeatureX&Feature2	Unspecified
DeepITE	Feature19	RootCause2	Feature60	RootCause3	Feature60&Feature19	RootCause2&RootCause3

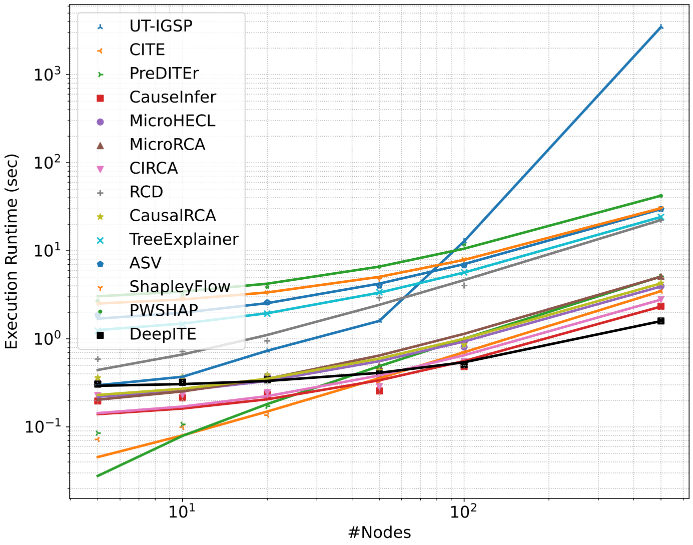


Figure 3: Runtime Analysis.

a given method identifies either feature X as an IT, it becomes challenging to ascertain whether RootCause 2 is indeed the true root cause.

Interestingly, DeepITE demonstrates superior performance on this dataset by effectively identifying features exclusive to a specific root cause. We evaluated the performance of all methods on samples 1758, 1760, and 1093, and summarized the results in Table 10. For sample 1758, where the true root cause is RootCause 2, DeepITE identifies feature19 as the IT, thus it is evident that RootCause2 should be the root cause. In contrast, other methods identify either feature X or feature Y, making it difficult to pinpoint the true root cause. Similarly, for sample 1760, DeepITE identifies feature60 as the IT, which is also exclusive to the true root cause, RootCause 3. On the other hand, in the case of sample 1093, DeepITE selects both feature60 and feature19, enabling us to conclude that both RootCause 2 and RootCause 3 are relevant root causes, a finding that aligns with the ground truth.

H Limitations

One limitation of the DeepITE framework is its applicability restricted to cases with fully observed causal graphs, presuming the absence of confounders. Real-world scenarios may involve confounding, where relationships between observed variables are influenced by latent variables. Addressing this challenge—how to effectively handle confounding in the presence of unobserved factors—represents a compelling avenue for future research. Additionally, DeepITE presupposes the availability of a pre-specified graph structure. While causal discovery techniques can be applied to ascertain the graph configuration when it is not known, the joint pursuit of determining both the graph structure and the intervention targets simultaneously offers a tantalizing challenge for future exploration. Finally, proving the consistency and identifiability of DeepITE, and more broadly in the application of VGAEs for ITE, remains an interesting avenue for future work. Notably, such theoretical guarantees have been established for VGAEs in the context of causal inference (both observational and interventional) in [56].

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have outline our contributions and scopes in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to Appendix H

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Please refer to Appendices C-E for the proofs of Propositions 1-3.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have well explained the proposed method in Section 5 with necessary derivations in Appendix F and further summarized the algorithm in Algorithm 1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our source code is available at `https://github.com/alipay/DeepITE`

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Please find the implementation details in Appendices G.1-G.4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All experiments report average results over 10 trials, with error bars representing a standard deviation ($\pm 1\sigma$) from the mean.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: As mentioned in Appendix G.1, all the training runs on 4 NVIDIA TESLA P100 GPUs with 50GB of VRAM. All the inference runs on a MacBook Pro 16 inch with a 6-core Intel i7 CPU and 16 GB of RAM.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have followed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This paper aims to contribute to the advancement of intervention target estimation, a fundamental research area not specific to any application. As such, we do not anticipate any direct societal impact resulting from our work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We will refrain from releasing any data or models with a high potential for misuse, as our research does not involve pretrained language models, image generators, or scraped datasets.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited all relevant papers.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our source code is available at `https://github.com/alipay/DeepITE`

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.