ROBUST ROOT CAUSE DIAGNOSIS USING IN-DISTRIBUTION INTERVENTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

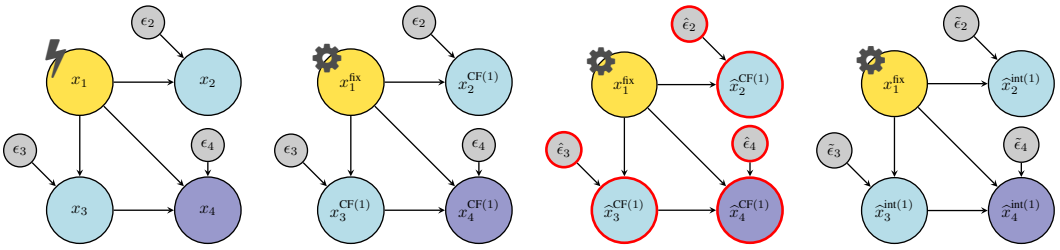
Diagnosing the root cause of an anomaly in a complex interconnected system is a pressing problem in today's cloud services and industrial operations. Effective root cause diagnosis calls for identifying nodes whose disrupted local mechanisms *cause* anomalous behavior at a target node. We propose In-Distribution Interventions (IDI), a novel algorithm that predicts root cause as nodes that meet two criteria: 1) *Anomaly:* root cause nodes should take on anomalous values; 2) *Fix*: had the root cause nodes assumed usual values, the target node would not have been anomalous. Prior methods of assessing the fix condition rely on counterfactuals inferred from a Structural Causal Model (SCM) trained on historical data. But since anomalies are rare and fall outside the training distribution, the fitted SCMs yield unreliable counterfactual estimates. IDI overcomes this by relying on interventional estimates obtained by solely probing the fitted SCM at in-distribution inputs. Our theoretical analysis demonstrates that IDI's in-distribution intervention approach outperforms other counterfactual estimation methods whenever variance of the underlying latent exogenous variables is low. Experiments on both synthetic and Petshop RCD benchmark datasets demonstrate that IDI consistently identifies true root causes more accurately and robustly than nine existing state-of-the-art RCD baselines. We release the anonymized code at `https://anonymous.4open.science/r/petshop-BB8A/`.

1 INTRODUCTION

In recent years, cloud services have gained popularity due to benefits such as resource sharing, scalability, and cost efficiency (Newman, 2021). Cloud-based systems consist of multiple interconnected nodes operating over a complex topology (Ashok et al., 2024; Hardt et al., 2023; Gu et al., 2024). Given the inherent complexity of such environments, faults are inevitable. These faults often manifest as anomalies, which represent significant deviations from usual behavior and are considered extremely low-probability events (Lomio et al., 2020). Anomalies can propagate through the system, affecting neighboring nodes and potentially compromising the entire application. Therefore, timely diagnosis of the *root cause* of anomalies is critical to minimize downtime and reduce operational costs. Cloud systems continuously monitor key performance indicators (KPIs) for nodes, including metrics like node latency, request counts, CPU utilization, and disk I/O (Meng et al., 2020). Anomalies are flagged at nodes when the KPIs associated with them deviate from expected patterns. However, automated root cause diagnosis (RCD) remains a challenge (Chen et al., 2019a).

We define root cause as nodes that satisfy two key conditions: (1) **Anomaly condition:** the root cause node exhibits anomalous behavior even while its causal parents are operating usually; and (2) **Fix condition:** had the root cause nodes assumed their usual values, the anomaly at the target node would not have occurred. While many prior RCD methods address the anomaly condition (Chen et al., 2014; Lin et al., 2018; Liu et al., 2021; Li et al., 2022), the fix condition is often overlooked, with some notable exceptions (Budhathoki et al., 2022b; Okati et al., 2024; Budhathoki et al., 2022a) that assess it via counterfactuals. Counterfactual estimation relies on learning a Structural Causal Model (SCM) (see Sec. 3.1). Given a causal graph linking the system's nodes, an SCM learns a set of functional equations that model the generation of each node as a function of its causal parents and latent exogenous variables. Such causal graphs can be derived from inverted call graphs that are easily available for cloud deployments Hardt et al. (2023).

A key challenge with SCMs is that they are typically trained on historical data collected under usual operating conditions. As a result, while SCMs can produce accurate counterfactuals for



(a) Root Cause sample (b) True Counterfactual (c) Est. Counterfactual (d) Est. Intervention

Figure 1: This figure demonstrates how abduction errors in counterfactuals impact RCD performance. In panel (a), an instance $\boldsymbol{x}$ shows an anomaly at the purple target node x_4, with the root cause being the gold node x_1, which is affected by an abnormal intervention. Hence x_1 takes OOD values, and influences its downstream nodes also to take on OOD values. Latent exogenous nodes are shown in grey. Panel (b) illustrates the true CF $\boldsymbol{x}^{\text{CF}(1)}$ obtained by applying the fix. Panel (c) presents the estimated CF for the fix, using exogenous *estimates* $\hat{\epsilon}$ – involving abduction that requires SCM evaluations in OOD regions (Sec. 3.1). Finally, panel (d) shows the interventional estimate, which uses *sampled* $\tilde{\epsilon}$, yielding a resulting $\widehat{x}_4^{\text{int}(1)}$ that conforms to the usual regime. Our theory in Sec 4 captures this rigorously.

in-distribution data, they become unreliable for anomalies that lie outside the training distribution (Okati et al., 2024). Our work reveals that these reliability issues stem from the first step in counterfactual estimation – abduction, where the functional equations in the SCM need to be inverted to estimate exogenous variables.

Figure 1 illustrates why estimating counterfactuals for anomalies leads to OOD evaluations. If the oracle values of latent exogenous variables were available, counterfactuals would be an ideal method for RCD. However, with causal mechanisms trained on data skewed towards usual behavior, the estimated exogenous variables incur significant error compared to the oracle. To overcome this, we introduce IDI (**I**n-**D**istribution **I**nterventions), which evaluates the fix condition through in-distribution interventions. A key feature of IDI is that for assessing the fix condition applied to true root cause, it needs inference of the SCM solely at its in-distribution regions, leading to a robust diagnosis. Our theoretical analysis compares IDI with counterfactual methods, showing that the latter's error scales with the total variational distance between the usual training and the rare anomalous distributions. In contrast, IDI's error is bounded by the standard deviation of latent exogenous variables, making IDI a superior choice except when exogenous variables exhibit high variance.

Our contributions in this work are:

1. We show, both theoretically and empirically, that SCMs trained on historical data suffer significant errors in the abduction phase for anomalies, rendering counterfactual-based approaches ineffective for RCD.
2. We propose IDI, an intervention-based approach that identifies root causes using in-distribution evaluations of the fitted SCM. This results in a lower error compared to counterfactual-based methods, except when the variance of exogenous variables is substantially high.
3. Experiments using cloud-based synthetic SCMs and a widely used RCD benchmark dataset with known causal graph (PetShop) demonstrate that IDI achieves greater robustness and accuracy in RCD compared to eleven baselines.

2 Related Work

Several anomaly detection methods (Akoglu, 2021; Chandola et al., 2009) use an anomaly scoring function g and a threshold τ to detect anomalies when an instance $\boldsymbol{x}$ has $g(\boldsymbol{x}) > \tau$. Our focus in this work is on RCD, and we group existing RCD approaches into Correlation and causal methods.

Correlation Approaches. Non-causal RCD methods typically rely on correlation-based analyses (Pham et al., 2024; cor, 2024; Chen et al., 2019b; Zhang et al., 1996; Chen et al., 2022; Luo et al., 2014; Ma et al., 2020; 2019; Yu et al., 2023) to assess relationships between a candidate root cause and the anomalous target node. Sometimes, spurious correlations make these methods predicting nodes that are not even causal ancestors of the target, leading to misleading conclusions.

Causal Approaches:. Causal methods use a causal graph $\mathcal{G}$ to limit root cause search to ancestors. Some methods propose learning the graph $\hat{\mathcal{G}}$, and may yield better results than the Correlation ones. In cloud deployments, it is best to leverage the easily available call graphs as learning a causal graph from training dataset requires some strong, and untestable assumptions (Glymour et al., 2019). **Causal Anomaly Approaches:** Some methods appraoch RCD using just the anomaly condition to identify nodes with disrupted local causal mechanisms (Chen et al., 2014; Lin et al., 2018; Liu et al., 2021; He et al., 2022; Meng et al., 2020; Xin et al., 2023; Yang et al., 2022; Okati et al., 2024; Yu et al., 2021; Wang et al., 2018; Li et al., 2022; Shan et al., 2019). They predict nodes x_j with low empirical sampling probabilities $P^{\text{trn}}_{X_j}(x_j|\text{Pa}_{x_j})$ conditioned on their causal parents as root causes. Some methods (Budhathoki et al., 2022a) attribute such abnormalities to latent exogenous disturbances, and they aim to infer abnormalities in the latent variables ϵ_j. **Causal Fix Approaches:** Sometimes, an abnormal ancestor may have a negligible causal effect on the target, disqualifying it as a root cause. The fix condition becomes necessary to disregard such nodes. Prior methods assess the fix condition via soft interventions (Jaber et al., 2020; Ikram et al., 2022), interventions (Okati et al., 2024), and counterfactuals (Budhathoki et al., 2022b;a). SAGE (Gan et al., 2021) is a prominent method in this category that leverages conditional variational autoencoders to model SCMs and estimates counterfactuals by sampling from the distribution generated by the encoder. However, in the RCD context, the encoder itself is conditioned on OOD inputs. Most methods probe their trained models with OOD inputs. In contrast, our approach, IDI, aims for RCD by probing its trained SCM on in-distribution inputs, thereby leading to a robust RCD approach.

3 Problem Formulation

3.1 Preliminaries

Notation. Let $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$ represent the n random variables denoting the Key Performance Indicators (KPIs) for each node in a system. We use $\boldsymbol{x} = (x_1, \ldots, x_n)$ to denote their realizations. These nodes are interconnected via a topology defined by a graph $\mathcal{G} = (\mathbf{X}, \mathbf{E})$ that is assumed to be directed and acyclic. An edge $(X_i, X_j) \in \mathbf{E}$ indicates that changes to X_i affect the values X_j can take, but not vice versa. We denote the parents of node i in the graph $\mathcal{G}$ as Pa_{X_i}, and the corresponding values assigned to these parents in $\boldsymbol{x}$ as Pa_{x_i}. We assume that each observed instance x_i is generated from its causal parents and a node specific latent exogenous variable ϵ_i via a structural causal model (SCM) as outlined below.

Structural Causal Models (SCM). An SCM (Pearl, 2019) is a four-tuple $\mathcal{S} = (\mathbf{X}, \boldsymbol{\epsilon}, \mathcal{F}, P_{\boldsymbol{\epsilon}})$, where $P_{\boldsymbol{\epsilon}}$ represents the distribution from which the exogenous variables are sampled. The observed variables $X_i \in \mathbf{X}$ are determined by a set of structural equations $\mathcal{F} = \{f_1, \ldots, f_n\}$, where $f_i : \text{Pa}_{X_i} \times \epsilon_i \mapsto X_i$. This implies that, conditioned on its immediate parents, no other variable can exert a causal influence on X_i—a principle known as *modularity* in causal inference. In summary, an SCM models a data-generating process where an observed instance $\boldsymbol{x}$ is produced by sampling the latent exogenous variables $\boldsymbol{\epsilon} \sim P_{\boldsymbol{\epsilon}}$ first, and then subsequently computing the node values by applying the structural equations in a topological order defined over the causal graph $\mathcal{G}$.

Counterfactuals. Given an observed instance $\boldsymbol{x} = (x_1, \ldots, x_n)$, we may want to generate a hypothetical instance $\boldsymbol{x}^{\text{CF}(j)}$ representing what $\boldsymbol{x}$ would have been if X_j had taken the value x'_j instead of x_j. This hypothetical instance, called a counterfactual, is computed using an SCM $\mathcal{S}$ in three steps:

1. *Abduction*: For each i, estimate $\hat{\epsilon}_i$ by inverting the function f_i so that $x_i = f_i(\text{Pa}_{x_i}, \epsilon_i)$ holds in the SCM $\mathcal{S}$.
2. *Action*: Set $x_i^{\text{CF}(j)} = x_i$ for any X_i that is not a descendant of X_j in $\mathcal{G}$. For X_j, set $x_j^{\text{CF}(j)} = x'_j$.
3. *Prediction*: For each descendant X_i of X_j, in topological order, set $x_i^{\text{CF}(j)} = f_i(\text{pa}_{x_i}^{\text{CF(j)}}, \hat{\epsilon}_i)$, using $\hat{\epsilon}_i$ obtained during the abduction step.

Interventions. An intervention $\boldsymbol{x}^{\text{int}(j)}$ represents the distributional effect of setting X_j to x'_j on its descendants in the causal graph $\mathcal{G}$. They are sampled as as follows:

1. Sample $\tilde{\epsilon}_i$ from its marginal distribution P_{ϵ_i}.

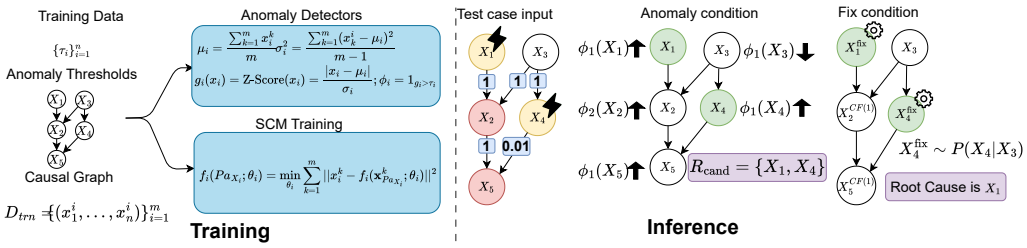


Figure 2: IDI's pipeline: During training, we use samples D_{trn}, a causal graph, and anomaly thresholds τ_i to learn anomaly detectors ϕ_i, and structural equations f_i as part of SCM training. During inference, we are given a root cause test case. The example illustrated two nodes X_1, X_4 with abnormal interventions. The edge weights indicate the strength of parent's influence (e.g., $\hat{f}_5(x_2, x_4) = x_2 + 0.04x_4$). while applying anomaly condition IDI discard X_2 in $\mathcal{R}_{\text{cand}}$ because of its anomalous parent X_1. Finally, the fix conditions excludes X_4 because it is insufficient to restore X_5 to a usual value. IDI declares X_1 as the root cause.

2. Steps 2 and 3 are the same as for counterfactuals, except that interventions use the sampled value $\tilde{\epsilon}_i$ in place of the abducted $\hat{\epsilon}_i$ during prediction; i.e., $\boldsymbol{x}_i^{\text{int}(j)} = f_i(\text{pa}_{x_i}^{\text{int(j)}}, \tilde{\epsilon}_i)$.

Note that the abduction step in counterfactuals necessitates each f_i to be invertible with respect to ϵ_i, whereas interventions do not need this requirement. Further, while $\boldsymbol{x}^{\text{CF}(j)}$ is a point estimate, $\boldsymbol{x}^{\text{int}(j)}$ is a random variable due to the randomness in $\tilde{\epsilon}_i$.

3.2 Training Setup

Our training dataset comprises the causal graph $\mathcal{G}$, and a set of N samples $\{\boldsymbol{x}^i\}_{i=1}^N$, where each $\boldsymbol{x}^i \in \mathbb{R}^n$. These samples are assumed to be drawn from an oracle SCM, $\mathcal{S} = (\mathbf{X} = \{X_1, \ldots, X_n\}, \boldsymbol{\epsilon} = \{\epsilon_1, \ldots, \epsilon_n\}, \mathcal{F} = \{f_1, \ldots, f_n\}, P_{\boldsymbol{\epsilon}}^{\text{trn}})$. Our training dataset D_{trn} encompasses samples that are obtained as follows from the Oracle SCM 1) sample $\boldsymbol{\epsilon}^i$ i.i.d. from $P_{\boldsymbol{\epsilon}}^{\text{trn}}$, and 2) compute $\boldsymbol{x}^i$ using the structural equations $\mathcal{F}$. Thus, most of the samples in D_{trn} exhibit usual behavior. To assess the anomaly level of each node $x_i \in \boldsymbol{x}$, we train a set of anomaly detectors $\varphi = \{(g_1, \tau_1), (g_2, \tau_2), \ldots, (g_n, \tau_n)\}$. Each detector φ_i consists of a scoring function $g_i : X_i \mapsto \mathbb{R}^+$ and an anomaly threshold $\tau_i > 0$. Thus, $\varphi_i(x_i) = 1_{g_i(x_i) > \tau_i}$ is the binary indicator for x_i being anomalous. These detectors can be trained unsupervised on D_{trn} using algorithms such as Z-score (Eidleman, 1995), isolation forest (Liu et al., 2008), or IT score (Budhathoki et al., 2022b). Given a test instance $\boldsymbol{x} = (x_1, \ldots, x_n)$ where an anomaly is detected at x_n, our **goal** is to trace the root cause of this anomaly to nodes in the system that caused it.

3.3 Qualifying Criteria for Root Cause

For ease of exposition, let us assume there is a unique root cause x_j, and establish the criteria that X_j needs to meet to qualify as a root cause. First, X_j must be actionable; i.e., these must exist a fix value x_j^{fix} such that applying it to X_j avoids anomaly at x_n. Furthermore, the abnormality at the root cause node should not have originated from any of its ancestors; otherwise, resolving the anomaly at x_n may also require fixing other nodes upstream of x_j. More formally, a root cause should satisfy the following two criteria:

1. *Anomaly Condition:* x_j must be anomalous, while its parent nodes are not, i.e., $\varphi_j(x_j) = 1$ and $\varphi_p(x_p) = 0$ for any parent node $p \in \text{Pa}_{X_j}$.
2. *Fix Condition:* Setting X_j to its fix value x_j^{fix} should have resolved the anomaly x_n. This implies the counterfactual $\boldsymbol{x}^{\text{CF}(j)}$, obtained by intervening on $X_j = x_j^{\text{fix}}$, should exhibit usual behavior at X_n; i.e., $\varphi_n(x_n^{\text{CF}(j)}) = 0$.

Defining the Root Cause Distribution. We define the distribution $Q_X^{\text{RC}(j)}$ that governs unique root cause samples $\boldsymbol{x}$, where x_j is the root cause of anomaly at x_n. Such a distribution will have $\boldsymbol{\epsilon}_{-j}$, representing all exogenous variables except ϵ_j, drawn from their usual distribution. Whereas, ϵ_j must be sampled strong enough to induce an anomaly at both X_n, X_j.

Definition 1 *We define the anomalous distribution $Q_X^{RC(j)}$ for unique root cause at X_j as:*

$$Q_{\boldsymbol{\epsilon}}^{RC(j)}(\boldsymbol{\epsilon}) = Q_{\boldsymbol{\epsilon}}^{RC(j)}(\boldsymbol{\epsilon}_{-j}) Q_{\boldsymbol{\epsilon}}^{RC(j)}(\epsilon_j | \boldsymbol{\epsilon}_{-j}) \tag{1}$$

These two factors are defined as:

$$Q_{\boldsymbol{\epsilon}}^{RC(j)}(\boldsymbol{\epsilon}_{-j}) = P_{\boldsymbol{\epsilon}}^{trn}(\epsilon_{-j}) \tag{2}$$

$$Q_{\boldsymbol{\epsilon}}^{RC(j)}(\epsilon_j | \boldsymbol{\epsilon}_{-j}) = P_{\boldsymbol{\epsilon}}^{trn}(\epsilon_j | \boldsymbol{\epsilon}_{-j}, \varphi_j(x_j) = 1, \varphi_n(x_n) = 1) \tag{3}$$

We denote the distribution induced by $Q_{\boldsymbol{\epsilon}}^{RC(j)}$ on the observed X through the SCM $\mathcal{S}$ as $Q_X^{RC(j)}$.

One key question regarding the fix condition is how to derive the value x_j^{fix}. We sample x_j^{fix} from its conditional distribution $P_X^{\text{trn}}(X_j \mid \text{Pa}_{x_j})$, mirroring how humans adjust x_j in practice. Moreover, this approach aligns with prior research (Budhathoki et al., 2022b) of sampling an exogenous fix ϵ_j^{fix} so as to induce usual values in X_j as a consequence.

The next question concerns how to estimate the causal effect of propagating the fix x_j^{fix} downstream to the target node X_n. We explore two approaches: the counterfactual estimate $\widehat{x}_n^{\text{CF}(j)}$ from Budhathoki et al. (2022b), and an alternative interventional estimate $\widehat{x}_n^{\text{int}(j)}$. Next, we conduct a theoretical analysis to highlight the limitations of both methods.

4 Interventions vs. Counterfactuals for RCD

We begin with the definitions used in our theoretical analysis.

Definition 2 *For two distributions P, Q defined over a space $\mathcal{X}$, the total variational distance (Redko et al., 2019) between them is defined as: $tvd(P, Q) = \frac{1}{2}\int_{x \in \mathcal{X}} |P(x) - Q(x)| dx$.*

Definition 3 *We say that an SCM $\mathcal{S}$ is an additive noise model when the structural equations are of the form $x_i = f_i(\text{Pa}_{X_i}) + \epsilon_i$, where f_i is a deterministic function, and ϵ_i is the exogenous variable.*

Let $\hat{P}_{\epsilon_i}^{\text{trn}}$ represent the estimated latent exogenous distribution $P_{\epsilon_i}^{\text{trn}}$, obtained from a validation dataset D_V. For additive noise models, $\hat{P}_{\epsilon_i}^{\text{trn}}$ is simply the empirical distribution of the error residuals $x_i - \hat{f}_i(\text{Pa}_{x_i})$ computed for each $\boldsymbol{x} \in D_V$. We need to use validation set to avoid the overfitting bias in estimation, that otherwise arises from using the train set (Chernozhukov et al., 2017).

Definition 4 *Consider an additive noise model $\mathcal{S}$, and let $\widehat{\mathcal{S}}$ be its estimate learned from training data $(D_{trn}, \mathcal{G})$. For a fix x_j^{fix} applied to the node X_j, let $\boldsymbol{x}^{CF(j)}$ be the true counterfactual from $\mathcal{S}$, and $\widehat{\boldsymbol{x}}^{CF(j)}, \widehat{\boldsymbol{x}}^{int(j)}$ be the estimated counterfactual and intervention from $\widehat{\mathcal{S}}$. Then, the following equations show how these quantities are derived:*

$$x_{j+1}^{CF(j)} = f_{j+1}(x_j^{fix}) + \epsilon_{j+1} \text{ where } \epsilon_{j+1} = x_{j+1} - f_{j+1}(x_j) \tag{4}$$

$$\widehat{x}_{j+1}^{CF(j)} = \hat{f}_{j+1}(x_j^{fix}) + \hat{\epsilon}_{j+1} \text{ where } \hat{\epsilon}_{j+1} = x_{j+1} - \hat{f}_{j+1}(x_j) \tag{5}$$

$$\widehat{x}_{j+1}^{int(j)} = \hat{f}_{j+1}(x_j^{fix}) + \tilde{\epsilon}_{j+1} \text{ where } \tilde{\epsilon}_{j+1} \sim \hat{P}_{\epsilon_{j+1}} \tag{6}$$

This procedure iterates for $j+2, \ldots, n$ in a topological order.

Since assessing the fix relies on $\varphi_n(x_n^{\text{CF}(j)})$, we bound the error incurred in estimating the true $x_n^{\text{CF}(j)}$ derived from the Oracle SCM $\mathcal{S}$ using $\widehat{x}_n^{\text{CF}(j)}$ from a fitted SCM $\widehat{\mathcal{S}}$.

Theorem 5 *Suppose the oracle SCM $\mathcal{S}$ is an additive noise model over a chain graph $\mathcal{G} = X_1 \to \cdots \to X_n$, with structural equations of the form $f_i(x_{i-1}) + \epsilon_i$, where each ϵ_i has bounded variance σ^2, and each function f_i is K-Lipschitz. Consider the hypothesis class $\mathcal{H} = \{\mathcal{H}_i\}_{i=1}^n$, where each $\mathcal{H}_i$ comprises bounded K-Lipschitz functions, resulting in losses bouned by $M > 0$. Let $\widehat{\mathcal{S}}$ be the SCM fitted on training data D_{trn}, with*

estimated functions $\{\hat{f}_i\}_{i=1}^n$. *Then, for test samples* $\boldsymbol{x}$ *drawn from the root cause distribution* $Q_X^{RC(j)}$, *with* X_j *as the unique root cause; for a fix* $x_j^{fix} \sim \hat{P}_X^{trn}(X_j|x_{j-1})$ *sampled from its empirical distribution, the estimated counterfactual* $\widehat{x}_n^{CF(j)}$ *computed from* $\widehat{\mathcal{S}}$ *satisfies:*

$$\mathbb{E}_{\boldsymbol{x}\sim Q_X^{RC(j)}}[|x_n^{CF(j)} - \widehat{x}_n^{CF(j)}|] \leq \sum_{i>j} K^{n-i}\Bigg[2^{n-i+1}\mathbb{E}_{x_{i-1}\sim P_{X_{i-1}}^{trn}}\left[|f_i(x_{i-1}) - \hat{f}_i(x_{i-1})|\right] + M^{n-i+1}\cdot\left[tvd\left(P_{X_{i-1}}^{trn}, Q_{X_{i-1}}^{RC(j)}\right) + \left(tvd\left(P_{X_{i-1}}^{trn}, P_{X_{i-1}}^{\hat{trn}}\right)\right)\right]\Bigg]$$

Proof Sketch: The main issue stems from abduction in Eq. 9, where $\hat{f}_{j+1}$ is inferred at an OOD input x_j. This introduces tvd(P, Q) terms in the bound. The exogenous error at a node propagates downstream to their descendants, further compounding the error at the target node. Please refer Fig. 1 for an illustration.

Remark: We first discuss the tvd$(P_{X_{i-1}}^{\text{trn}}, Q_{X_{i-1}}^{\text{RC}(j)})$ terms that arise from *abduction*. In a geometric series, the leading term dominates the sum. Here, the leading term involves tvd$(P_{X_j}^{\text{trn}}, Q_{X_j}^{\text{RC}(j)})$, where $\varphi_j(x_j) = 1$. Thus, for the root cause X_j the above discrepancy lies between the usual training distribution and the anomalous test distribution. Since this discrepancy is substantial, $\widehat{x}_n^{\text{CF}(j)}$ is a poor estimate of $x_n^{\text{CF}(j)}$. Other terms in the bound reflect discrepancies between i.i.d. train, test errors, and can be shown to reduce with increasing training size $|D_{\text{trn}}|$ using generalization bounds like VC-dimensions, etc. (Redko et al., 2019).

Next, we conduct a similar analysis to assess the error in using $\widehat{x}_n^{\text{int}(j)}$ as an estimate for $x_n^{\text{CF}(j)}$.

Theorem 6 *Under the same conditions laid out in Theorem 5, the error between the true counterfactual* $x_n^{CF(j)}$ *and the estimated intervention* $\widehat{x}_n^{int(j)}$ *admits the following bound:*

$$\mathbb{E}_{\boldsymbol{x}\sim Q_X^{RC(j)}}\left[|x_n^{CF(j)} - \widehat{x}_n^{int(j)}|\right] \leq \sum_{i>j} K^{n-i}\Bigg[\mathbb{E}_{x_{i-1}\sim P_{X_{i-1}}^{trn}}\left[|f_i(x_{i-1}^{CF(j)}) - \hat{f}_i(x_{i-1}^{CF(j)})|\right] + M^{n-i+1}\cdot\left(tvd\left(P_{X_{i-1}}^{trn}, P_{X_{i-1}}^{\hat{trn}}\right)\right) + std(\epsilon_i) + std(\tilde{\epsilon}_i) + |\mathbb{E}[\epsilon_i] - \mathbb{E}[\tilde{\epsilon}_i]|\Bigg]$$

where $std(\bullet)$ *denotes the standard deviation.*

Proof Sketch: Unlike in the CF case, the estimation of $\widehat{x}_n^{\text{int}(j)}$ does not need abduction. Instead, it samples $\tilde{\epsilon}_{j+1}$ as shown in Eq. 6, and the error from abduction is limited by the standard deviation of the oracle ϵ_j. A key advantage of this is that the challenging tvd(P, Q) terms are eliminated from the bound, and these samples $\tilde{\epsilon}_{j+1}$ result in only in-distribution inputs to the fitted SCM.

Corollary 7 *Suppose in Theorem 6, the exogenous variables* ϵ_i *are zero-mean in addition to having bounded variance for any* $i \in [n]$, *then the two terms* $std(\tilde{\epsilon}_i)$ *and* $|\mathbb{E}[\epsilon_i] - \mathbb{E}[\tilde{\epsilon}_i]|$ *can be dropped from the bound in Theorem 6.*

Remark: All terms, except std(ϵ_i), can be simplified using generalization bounds. Importantly, interventional estimates remain stable despite the drift between the training distribution P_X^{trn} and the anomalous distribution $Q_X^{\text{RC}(j)}$. Thus, when exogenous variables have low variance, interventions provide a more robust method for estimating $x_n^{\text{CF}(j)}$ and evaluating the fix condition.

We defer all proofs to Appendix B.

5 Our Approach: IDI

IDI uses interventions motivated by the analysis in Sec. 4 as they offer a robust approach to RCD. We illustrate the training and inference procedure of IDI in Fig. 2. IDI first applies the anomaly condition to filter the promising root cause candidates into a set $\mathcal{R}_{\text{cand}}$. Then, it applies the fix condition to nodes in $\mathcal{R}_{\text{cand}}$ to diagnose the root cause. We lay down the procedure for unique root cause and multiple root cause scenarios separately. For multiple root cause diagnosis, we need the following assumption to be met:

Assumption 1 *At most one root cause exists in every simple path that leads to the target X_n in $\mathcal{G}$.*

Step 1 of IDI: Assessing the anomaly condition. The goal of this step is to identify nodes whose exogenous variables are abnormal. IDI's approach is straightforward; it iterates over all ancestors of X_n in $\mathcal{G}$, adding node X_i to $\mathcal{R}_{\text{cand}}$ if X_i shows an anomaly but none of its parents do. Theorem 4.5 in (Okati et al., 2024) proves that this approach is sound for chain graphs. To assess anomalies, we use the Z-Score, defined for X_i as Z-score$(x_i) = \frac{|x_i - \mu_i|}{\sigma_i}$ where μ_i and σ_i are the sample mean and standard deviation computed for the ith node in the training data. Several other methods have been proposed for anomaly criterion (Chen et al., 2014; Li et al., 2022; Okati et al., 2024), and IDI can incorporate any of them.

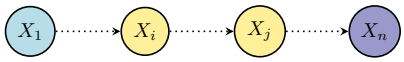


Figure 3: A chain graph with more than one root cause in a simple path to X_n. Dotted lines denote a directed path.

Next, We illustrate the need for assumption 1 using a chain graph in Fig. 3 that comprises two root cause nodes, X_i and X_j, along a simple directed path to X_n. Since X_i is a root cause, it is anomalous and is likely to influence its downstream nodes to take on anomalous values, including X_j's parent. As a result, IDI may discard X_j incorrectly.

Step 2 of IDI: Assessing the fix Condition. We first describe the procedure for the simpler of unique root cause, and then generalize to multiple root causes.

Unique root cause: In this case, IDI applies the fix condition on nodes in $\mathcal{R}_{\text{cand}}$ iteratively. Suppose X_j is the true cause, our fix value x_j^{fix} applied to X_j suppresses the only abnormal ϵ_j that caused the anomaly x_n. Since IDI samples all other exogenous variables downstream of X_j from their usual distributions, the fix condition for true root cause is assessed by probing the learned SCM at in-distribution inputs. Therefore, $\varphi_n(\widehat{x}_n^{\text{int}(j)})$ would evaluate to 0 and IDI correctly predicts X_j.

Multiple root causes: In this case, we require set-valued fixes to avoid OOD evaluations. We justify this need with an example in Fig. 4. Let $\alpha^\star = \{X_i, X_j\}$ be two root cause nodes. Since they intersect at a common descendant X_k, a fix applied to X_i leads to an OOD evaluation at X_k due to the influence of its anomalous ancestor X_j. This can potentially causing errors in $\widehat{x}_k^{\text{int}(i)}$ inference that propagate to $\widehat{x}_n^{\text{int}(i)}$. However, when both X_i and X_j are fixed simultaneously, all evaluations will be in-distribution.

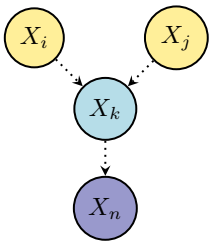


Figure 4: An example graph with more than one root cause.

Another issue with interventions is that any superset of $\alpha^\star$ may appear to resolve the anomaly, leading to over-prediction of root causes. For instance, applying a fix to $\alpha \subset \alpha^\star$ suppresses the abnormalities in both the root causes in $\alpha^\star$ and also some redundant nodes in $\alpha^\star \setminus \alpha$. To address this, we leverage Shapley values (Shapley, 1953) from cooperative game theory. Given a set $\mathcal{U}$ of m items and a utility function $\phi : 2^{\mathcal{U}} \mapsto \mathbb{R}^+$ that assigns a score to each subset $\alpha \subseteq \mathcal{U}$, Shapley values provide a fair method to distribute credit among the m players. A player i that achieves a high score of $\phi(\alpha \cup \{i\}) - \phi(\alpha)$ across multiple $\alpha \subset \mathcal{U}$ receives a high Shapley value. We treat $\mathcal{R}_{\text{cand}}$ as the items and define the utility function for $\alpha \subseteq \mathcal{R}_{\text{cand}}$ as the reduction in raw anomaly scores at X_n after fixing α: $g_n(x_n) - g_n(\widehat{x}_n^{\text{int}(\alpha)})$. Any subset that includes $\alpha^\star$ reduces the raw anomaly scores significantly thereby achieving high Shapley values. The pseudocode for the multi-root-cause diagnosis algorithm is presented in Alg 1 in the Appendix.

6 Experiments

We conduct experiments on petshop, a benchmark deployed cloud RCD dataset with known call graph, and several synthetic datasets. We compare IDI against following baselines:

- **Correlation** This class includes `ϵ-diagnosis` (Shan et al., 2019), which identifies root causes by testing for significant behavior changes during anomalies, and `ranked correlation` (Hardt et al., 2023), which assigns root cause to nodes correlated with the target.
- **Causal Anomaly** This class encompasses `traversal` methods (Chen et al., 2014; Lin et al., 2018; Liu et al., 2021; Meng et al., 2020), broadly grouped by (Okati et al., 2024), including the `smooth traversal` method introduced by them. These methods implement a version our

anomaly condition. `CIRCA` (Li et al., 2022) identifies nodes connected to the target through fully anomalous paths, while `random walk` methods (Yu et al., 2021) use heuristics.

- **Causal Fix** This includes Hierarchical RCD (`HRCD`) (Ikram et al., 2022), which predicts root causes as nodes that suffered local mechanism changes affecting the target. `TOCA` (Okati et al., 2024) implements the fix condition jointly over all nodes. The `CF Attribution` (Budhathoki et al., 2022b) method uses Shapley values to perform CF contribution analysis on *all* n nodes.

We implemented IDI in the RCD library released by Petshop (Hardt et al., 2023)[1]. This library includes implementations for most baseline methods used in our study, and we implemented `TOCA`, `smooth traversal`. Petshop uses Dowhy (Sharma & Kiciman, 2020) and gcm (Blöbaum et al., 2022) for causal inference and PyRCA (Liu et al., 2023) for root cause analysis. To ensure fair comparisons, we applied the same experimental settings for IDI as those used for the baselines in Petshop.

Evaluation Metric. We assess root cause prediction accuracy using Recall@k (Ikram et al., 2022). Let $\alpha^\star \subset \mathbf{X}$ be the ground truth root causes, and $\hat{\alpha}$ the ordered list of predicted root causes, where $\hat{\alpha}[1]$ is the most prominent. For any $k > 0$

$$\text{Recall@k}(\hat{\alpha}, \alpha^\star) = \frac{\sum_{i=1}^{|\alpha^\star|+k-1} 1_{\hat{\alpha}[i] \in \alpha^\star}}{|\alpha^\star|}; \quad 1_{\hat{\alpha}[i] \in \alpha^\star} = 1 \text{ if } \hat{\alpha}[i] \in \alpha^\star \tag{7}$$

For $k = 1$, we see that Recall@$1(\hat{\alpha}, \alpha^\star) = 1$ iff every node in $\alpha^\star$ is present in $\hat{\alpha}[1:k]$, while for larger k, we need $\alpha^\star$ to be present in the first $|\alpha^\star| + k - 1$ predictions of $\hat{\alpha}$. We assess using $k = 1, 3$.

6.1 Experiments on Petshop: RCD in a Deployed Cloud System

		Low		High		Temporal	
Recall@		k=1	k=3	k=1	k=3	k=1	k=3
Correlation	Random Walk (Yu et al., 2021)	0.00	0.10	0.00	0.20	0.00	0.33
	Ranked Correlation (Hardt et al., 2023)	0.40	0.60	0.70	0.90	0.50	0.67
	ϵ-Diagnosis (Shan et al., 2019)	0.00	0.00	0.00	0.00	0.17	0.17
Causal Anomaly	Circa (Li et al., 2022)	0.60	0.80	0.60	1.00	0.67	1.00
	Traversal (Chen et al., 2014)	0.80	0.80	0.90	0.90	1.00	1.00
	Smooth Traversal (Okati et al., 2024)	0.40	0.60	0.00	0.60	0.50	1.00
Causal Fix	HRCD (Ikram et al., 2022)	0.07	0.21	0.00	0.07	0.25	0.75
	TOCA (Okati et al., 2024)	0.40	0.40	0.20	0.20	0.00	0.00
	CF Attribution (Budhathoki et al., 2022b)	0.40	0.60	0.40	0.70	0.00	0.50
	IDI (Ours)	0.90	0.90	0.90	0.90	1.00	1.00

Table 1: Diagnosing root causes of latency issues in petshop, a cloud-based microservices dataset. Methods are categorized into Correlation, Causal Anomaly, and Causal Fix.

Petshop (Hardt et al., 2023) is a recent dataset designed for benchmarking RCD methods in the cloud domain, featuring a call graph $\mathcal{G}$ that causally links key performance indicators (KPIs). The baseline methods in the Petshop library use a linear additive noise model $\widehat{\mathcal{S}}$, which we also adopted for IDI. This dataset encompasses three types of latency issues: low, high, and temporal, with many methods successfully identifying the temporal issues. Overall, IDI outperforms other methods in most settings, except for Recall@3 in high latency, where `CIRCA` emerged as the best performer. `CIRCA` identifies root causes as nodes connected to the target through all anomalous nodes, and possibly high latency test cases showed such a favorable behavior. Otherwise, `Traversal` proved to be a strong contender against IDI. The improvements observed in IDI across various settings can be attributed solely to its robust implementation of the fix condition. In contrast, the gains seen in other methods assessing the fix condition are less pronounced compared to Causal Anomaly methods, as their evaluations involve applying $\widehat{\mathcal{S}}$ to out-of-distribution inputs.

6.2 Experiments on Synthetic SCMs

Next, we design synthetic experiments to answer three key research questions:

RQ1 *Linear SCM:* How effective is RCD when the Oracle SCM $\mathcal{S}$ has linear structural equations, so that an $\hat{\mathcal{S}}$ fit using samples from the usual distribution also generalizes OOD?

[1] `https://github.com/amazon-science/petshop-root-cause-analysis`

RQ2 *Non-Lin Invertible SCM:* How effective is RCD when $\hat{\mathcal{S}}$ closely approximates a non-linear $\mathcal{S}$ within the usual regime, and $\mathcal{S}$ allows abduction, meaning f_i is invertible with respect to ϵ_i?

RQ3 *Non-Lin Non-Invertible SCM:* What are the implications for root cause identification when $\hat{\mathcal{S}}$ closely matches a non-linear $\mathcal{S}$ in the usual distribution, but $\mathcal{S}$ does not support abduction?

We evaluate each option under unique and multiple root cause scenarios. For multiple root causes, we ensure they follow the assumption 1, and later perform ablations under its violations in Table 3. Our synthetic setup involved a single anomalous test sample for RCD, so we did not run two baselines: 1) `ϵ-diagnosis` method, which requires multiple anomalous samples for conducting two-sample tests, and 2) `HRCD` which learns the causal graph solely from anomalous samples.

Generating the Oracle SCM $\mathcal{S}$: We randomly sample a causal graph $\mathcal{G}$ using the Networkx library Hagberg et al. (2008). We select the node with the most ancestors as the anomaly target, and the root cause nodes $\alpha^\star$ arbitrarily. Since cloud KPIs, such as node latency and CPU utilization, are typically positive (Meng et al., 2020), we ensured that all nodes in the synthetic data assume positive values. Exogenous variables follow a uniform distribution $\epsilon_i \sim U[0, 1]$ making their standard deviation $\text{std}(\epsilon_i) = 0.3$. We explore other choices for ϵ_i in Appendix E. Finally, We generate the training dataset D_{trn} by sampling exogenous variables $\boldsymbol{\epsilon}$, and then use $\mathcal{S}$ to generate the observed nodes $\boldsymbol{x}$ in a topological order. Each node in $\mathcal{G}$ is assigned a local causal mechanism as follows:

- For linear SCMs in RQ1, we define the functional equations as linear, with random coefficients as: $x_i = w_i^\top \text{Pa}_{x_i}$ where $w_i \sim U[0.5, 2]$.
- For non-lin invertible SCMs in RQ2, we set each local mechanism f_i as an additive noise three layer ELU activation based MLP.
- For non-lin non-invertible SCMs, we use the same MLP architecture, but without additive noise. In this case, each MLP f_i receives both the parents Pa_{x_i} and the noise ϵ_i as inputs.

For linear SCM, we fit a linear additive noise SCM $\widehat{\mathcal{S}}$, whereas for both RQ2 and RQ3, we fit an additive noise MLP-based SCM $\widehat{\mathcal{S}}$. We show some example graphs in Appendix G.

Test case generation: For the root cause set $\alpha^\star$ to cause an anomaly at x_n, we first sample $\boldsymbol{\epsilon}_{-\alpha^\star}$ from $U[0, 1]$, then apply a grid search over $\boldsymbol{\epsilon}_{\alpha^\star}$ so that $(\boldsymbol{\epsilon}_{\alpha^\star}, \boldsymbol{\epsilon}_{-\alpha^\star})$ together lead the SCM $\mathcal{S}$ to induce an anomaly at x_n. We also introduced some irrelevant anomalies at nodes that have weak functional relationships with x_n to assess the impact on methods that ignore the fix condition. We repeat each experiment 10 times and report the average values of Recall@$k \in \{1, 3\}$.

	RQ1 Linear				RQ2 Non-Lin Invertible				RQ3 Non-Lin Non-Invertible			
Number root causes	Unique		Multiple		Unique		Multiple		Unique		Multiple	
Recall@	$k=1$	$k=3$	$k=1$	$k=3$	$k=1$	$k=3$	$k=1$	$k=3$	$k=1$	$k=3$	$k=1$	$k=3$
Random Walk	0.00	0.00	0.03	0.03	0.10	0.30	0.27	0.27	0.10	0.30	0.27	0.27
Ranked Correlation	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Traversal	0.50	1.00	0.40	0.67	0.00	1.00	0.03	0.48	0.00	0.80	0.00	0.33
Smooth Traversal	0.50	1.00	0.52	0.90	0.50	1.00	0.57	0.83	0.40	0.80	0.40	0.73
CIRCA	0.80	1.00	0.33	0.33	0.60	0.80	0.38	0.48	0.40	0.70	0.03	0.03
TOCA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CF Attribution	1.00	1.00	0.97	1.00	0.30	0.70	0.20	0.40	0.00	0.20	0.00	0.23
IDI (ours)	1.00	1.00	1.00	1.00	1.00	1.00	0.83	0.97	0.60	0.80	0.63	0.83

Table 2: Recall@k=1 and 3 values for experiments on synthetic SCMs. We present the recall values averaged across ten runs. Best methods are highlighted in green, while the second best methods are shown in yellow. Overall, IDI demonstrates the highest recall across all settings.

Results: We report the results in Table 2. Correlation methods struggle across all settings because a root cause node induces an anomalous behavior across all its descendants, making them highly correlated with the target X_n. These methods fail to isolate the root. This highlights the need for detecting local *causal* mechanism changes for RCD. In linear SCMs, many methods successfully detect the unique root cause. When there are multiple root causes, the number of anomalous downstream nodes between them and the target increases significantly, leading to an increase in false positives. Among Causal Anomaly methods, `smooth traversal` performs best. The `CF Attribution` method excels with linear SCMs due to its ability to perform accurate abduction. However, its performance deteriorates significantly in non-linear scenarios. `TOCA` evaluates all nodes indiscriminately, resulting in numerous OOD evaluations of the fitted SCM, whereas IDI ef-

fectively scrutinizes just the $\mathcal{R}_{\text{cand}}$. Overall, IDI outperforms other methods by a substantial margin across various settings.

6.3 Experiments under Assumption 1 Violations

In this experiment, we injected root causes at arbitrary nodes, resulting in Assumption 1 violations. The results are shown in Table 3. Both IDI and other Causal Anomaly methods face challenges in this scenario as they need parents of a root cause node to be usual. While `CF attribution` performs best in the linear setting, it struggles in other settings due to abduction errors being amplified by the presence of *multiple* root causes in the same path. For non-lin inv SCMs, IDI achieves the highest Recall, while in non-lin non-inv cases, it surpasses the CF method by $2\times$ Recall. Overall, IDI achieved the best method Recall even under assumption 1 violations.

	Linear		Non-Lin Inv		Non-Lin Non-Inv	
	Top-1	Top-3	Top-1	Top-3	Top-1	Top-3
Random Walk	0.10	0.10	0.13	0.13	0.13	0.13
Ranked Correlation	0.00	0.00	0.00	0.00	0.00	0.00
Traversal	0.00	0.40	0.03	0.27	0.00	0.27
Smooth Traversal	0.23	0.50	0.17	0.47	0.30	0.43
CIRCA	0.13	0.13	0.27	0.27	0.13	0.13
TOCA	0.07	0.07	0.03	0.03	0.00	0.00
CF Attribution	0.83	0.97	0.33	0.57	0.07	0.23
IDI	0.57	0.57	0.53	0.60	0.40	0.53

Table 3: Experiment under Assumption 1 violations. A simple path to X_n can features more than one root cause.

6.4 Experiment on IDI vs. IDI (using CFs)

	Synthetic Oracle SCM						Petshop Latency		
	Linear		Non-Lin Invertible		Non-Lin Non Invertible		low	high	temporal
	Unique	Multiple	Unique	Multiple	Unique	Multiple			
CF Attribution	1.00	0.97	0.30	0.20	0.00	0.00	0.40	0.40	0.00
IDI (CF)	1.00	1.00	1.00	0.77	0.60	0.12	0.70	0.70	1.00
IDI	1.00	1.00	1.00	0.83	0.70	0.63	0.90	0.90	1.00

Table 4: Recall@1 for CF Attribution - as baseline method (Budhathoki et al., 2022b), IDI (CF) a version of IDI that uses CF in step-2, and IDI. IDI performs the best.

In this experiment, we run IDI in counterfactual (CF) mode, referred to as IDI (CF). The first step remains the same, filtering root cause candidates $\mathcal{R}_{\text{cand}}$, but in the second step, it applies Shapley analysis on estimated counterfactuals on $\mathcal{R}_{\text{cand}}$ instead of IDI's usual interventions. As a baseline, we also compare CF attribution method that skips $\mathcal{R}_{\text{cand}}$ filtering and runs CF Shapley analysis on all nodes. Results in Table 4 show that CF attribution performs worse than IDI (CF), confirming that filtering $\mathcal{R}_{\text{cand}}$ isolates promising candidates, leading to in-distribution evaluations of the fitted SCM. While IDI (CF) performs well in linear settings, its performance drops across other settings. This drop can solely be attributed to CFs, highlighting that interventions are a more effective approach for RCD.

7 Limitations and Conclusion

Limitations: (1) IDI's performance can degrade when Assumption 1 is violated. (2) In additive noise models, when the training size is large enough for the estimated SCM to approach the oracle SCM, errors in using interventional estimates of counterfactuals plateau at the std. deviation of the exogenous variables (Appendix E). Whereas, counterfactual estimates converge to the true values.

Conclusion: In this paper, we introduced two key conditions—Anomaly and Fix—to identify root causes of anomalies. While prior methods effectively addressed the anomaly condition, the fix condition often relied on probing trained models with OOD inputs. To address this, we proposed IDI, a novel in-distribution intervention method that ensures the fitted SCM is probed using in-distribution inputs while evaluating the potential of true root cause nodes to resolve the anomaly. Unlike previous methods that required a unique root cause assumption, IDI operates under the more relaxed condition of at most one cause per path. We showed theoretically that IDI's intervention method is superior to counterfactual approaches for additive noise chain SCMs. Our experiments with arbitrary SCMs reaffirmed IDI's capability to deliver robust and accurate RCD.

REFERENCES

Automated root cause analysis with watchdog rca, 2024. URL https://www.datadoghq.com/blog/datadog-watchdog-automated-root-cause-analysis/.

Leman Akoglu. Anomaly mining: Past, present and future. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 1–2, 2021.

Sachin Ashok, Vipul Harsh, Brighten Godfrey, Radhika Mittal, Srinivasan Parthasarathy, and Larisa Shwartz. Traceweaver: Distributed request tracing for microservices without application modification. In *Proceedings of the ACM SIGCOMM 2024 Conference*, ACM SIGCOMM '24, pp. 828–842, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400706141. doi: 10.1145/3651890.3672254. URL https://doi.org/10.1145/3651890.3672254.

Patrick Blöbaum, Peter Götz, Kailash Budhathoki, Atalanti A. Mastakouri, and Dominik Janzing. Dowhy-gcm: An extension of dowhy for causal inference in graphical causal models, 2022.

Kailash Budhathoki, George Michailidis, and Dominik Janzing. Explaining the root causes of unit-level changes, 2022a. URL https://arxiv.org/abs/2206.12986.

Kailash Budhathoki, Lenon Minorics, Patrick Bloebaum, and Dominik Janzing. Causal structure-based root cause analysis of outliers. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 2357–2369. PMLR, 17–23 Jul 2022b.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

Junjie Chen, Xiaoting He, Qingwei Lin, Yong Xu, Hongyu Zhang, Dan Hao, Feng Gao, Zhangwei Xu, Yingnong Dang, and Dongmei Zhang. An empirical investigation of incident triage for online service systems. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pp. 111–120. IEEE, 2019a.

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*, pp. 1887–1895, 2014. doi: 10.1109/INFOCOM.2014.6848128.

Yujun Chen, Xian Yang, Qingwei Lin, Hongyu Zhang, Feng Gao, Zhangwei Xu, Yingnong Dang, Dongmei Zhang, Hang Dong, Yong Xu, Hao Li, and Yu Kang. Outage prediction and diagnosis for cloud service systems. In *The World Wide Web Conference*, WWW '19, pp. 2659–2665, New York, NY, USA, 2019b. Association for Computing Machinery.

Zhuangbin Chen, Jinyang Liu, Yuxin Su, Hongyu Zhang, Xiao Ling, Yongqiang Yang, and Michael R Lyu. Adaptive performance anomaly detection for online service systems via pattern sketching. In *Proceedings of the 44th International Conference on Software Engineering*, pp. 61–72, 2022.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and causal parameters, 2017. URL https://arxiv.org/abs/1608.00060.

Gregory J Eidleman. Z scores-a guide to failure prediction. *The CPA Journal*, 65(2):52, 1995.

Yu Gan, Mingyu Liang, Sundar Dev, David Lo, and Christina Delimitrou. Sage: practical and scalable ml-driven performance debugging in microservices. ASPLOS '21, pp. 135–151, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383172. doi: 10.1145/3445814.3446700. URL https://doi.org/10.1145/3445814.3446700.

Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

Xuhang Gu, Qingyang Wang, Jianshu Liu, and Jinpeng Wei. Grunt attack: Exploiting execution dependencies in microservices. In *2024 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 115–128, 2024. doi: 10.1109/DSN58291.2024.00025.

Aric Hagberg, Pieter J Swart, and Daniel A Schult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2008.

Michaela Hardt, William Orchard, Patrick Blöbaum, Shiva Kasiviswanathan, and Elke Kirschbaum. The petshop dataset – finding causes of performance issues across microservices, 2023.

Zilong He, Pengfei Chen, Yu Luo, Qiuyu Yan, Hongyang Chen, Guangba Yu, and Fangyuan Li. Graph based incident extraction and diagnosis in large-scale online systems. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering (ASE'22)*, pp. 1–13, 2022.

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.

Amin Jaber, Murat Kocaoglu, Karthikeyan Shanmugam, and Elias Bareinboim. Causal discovery from soft interventions with unknown targets: Characterization and learning. In *Advances in Neural Information Processing Systems (NeurIPS'20)*, volume 33, pp. 9551–9561, 2020.

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*, pp. 3230–3240, 2022.

JinJin Lin, Pengfei Chen, and Zibin Zheng. Microscope: Pinpoint performance issues with causal graphs in micro-service environments. In *International Conference on Service Oriented Computing*, 2018. URL `https://api.semanticscholar.org/CorpusID:53233176`.

Chenghao Liu, Wenzhuo Yang, Himanshu Mittal, Manpreet Singh, Doyen Sahoo, and Steven CH Hoi. Pyrca: A library for metric-based root cause analysis. *arXiv preprint arXiv:2306.11417*, 2023.

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 *Proceedings of the 43rd International Conference on Software Engineering: Software Engineering in Practice*, ICSE-SEIP '21, pp. 338–347. IEEE Press, 2021. ISBN 9780738146690.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, 2008. doi: 10.1109/ICDM.2008.17.

Francesco Lomio, Diego Martínez Baselga, Sergio Moreschini, Heikki Huttunen, and Davide Taibi. Rare: a labeled dataset for cloud-native memory anomalies. In *Proceedings of the 4th ACM SIGSOFT International Workshop on Machine-Learning Techniques for Software-Quality Evaluation*, pp. 19–24, 2020.

Chen Luo, Jian-Guang Lou, Qingwei Lin, Qiang Fu, Rui Ding, Dongmei Zhang, and Zhe Wang. Correlating events with time series for incident diagnosis. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1583–1592, 2014.

Meng Ma, Weilan Lin, Disheng Pan, and Ping Wang. Ms-rank: Multi-metric and self-adaptive root cause diagnosis for microservice applications. In *IEEE International Conference on Web Services (ICWS'19)*, pp. 60–67, 2019.

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*, WWW'20, pp. 246–258, 2020.

Yuan Meng, Shenglin Zhang, Yongqian Sun, Ruru Zhang, Zhilong Hu, Yiyin Zhang, Chenyang Jia, Zhaogang Wang, and Dan Pei. Localizing failure root causes in a microservice through causality inference. In *2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*, pp. 1–10. IEEE, 2020.

Sam Newman. *Building microservices*. " O'Reilly Media, Inc.", 2021.

Nastaran Okati, Sergio Hernan Garrido Mejia, William Roy Orchard, Patrick Blöbaum, and Dominik Janzing. Root cause analysis of outliers with missing structural knowledge, 2024. URL `https://arxiv.org/abs/2406.05014`.

Judea Pearl. Sufficient causes: On oxygen, matches, and fires. *Journal of Causal Inference*, 7(2), 2019.

Luan Pham, Huong Ha, and Hongyu Zhang. Baro: Robust root cause analysis for microservices via multivariate bayesian online change point detection. *Proceedings of the ACM on Software Engineering*, 1(FSE):2214–2237, 2024.

Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younes Bennani. *Advances in domain adaptation theory*. Elsevier, 2019.

Huasong Shan, Yuan Chen, Haifeng Liu, Yunpeng Zhang, Xiao Xiao, Xiaofeng He, Min Li, and Wei Ding. ε-diagnosis: Unsupervised and real-time diagnosis of small-window long-tail latency in large-scale microservice platforms. In *The World Wide Web Conference*, pp. 3215–3222, 2019.

Lloyd S Shapley. A value for n-person games. *Contribution to the Theory of Games*, 2, 1953.

Amit Sharma and Emre Kiciman. Dowhy: An end-to-end library for causal inference, 2020.

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)*, pp. 492–502. IEEE, 2018.

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*, 203: 111724, 2023.

Wenzhuo Yang, Kun Zhang, and Steven CH Hoi. A causal approach to detecting multivariate time-series anomalies and root causes. *arXiv preprint arXiv:2206.15033*, 2022.

Guangba Yu, Pengfei Chen, Hongyang Chen, Zijie Guan, Zicheng Huang, Linxiao Jing, Tianjun Weng, Xinmeng Sun, and Xiaoyun Li. Microrank: End-to-end latency issue localization with extended spectrum analysis in microservice environments. In *Proceedings of the Web Conference 2021*, pp. 3087–3098, 2021.

Qingyang Yu, Changhua Pei, Bowen Hao, Mingjie Li, Zeyan Li, Shenglin Zhang, Xianglin Lu, Rui Wang, Jiaqi Li, Zhenyu Wu, et al. Cmdiagnostor: An ambiguity-aware root cause localization approach based on call metric data. In *Proceedings of the ACM Web Conference 2023*, pp. 2937–2947, 2023.

Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. *ACM sigmod record*, 25(2):103–114, 1996.

A IDI ALGORITHM

Here we provide the pseudocode for IDI.

Algorithm 1 IDI Algorithm

Require: Data D_{trn}, DAG $\mathcal{G}$, anomalous instance $\boldsymbol{x}$, anomaly config $\mathcal{A}$
Ensure $\widehat{\mathcal{R}}$ ◁ predicted root causes
$\{\varphi_i, g_i\} \leftarrow \texttt{TrainAnomalyFns}(D_{\text{trn}}, \mathcal{G}, \mathcal{A})$ ◁ $\mathcal{A}$ is part of problem spec; default g_i is Z-Score
$\mathcal{R}_{\text{cand}} \leftarrow \{i : \varphi_i(x_i) = 1, \text{ and } \varphi_p(x_p) = 0 \ \forall\, p \in \text{Pa}_{X_i}\}$ ◁ anomaly condition
$\widehat{\mathcal{S}} \leftarrow \text{FitSCM}(D_{\text{trn}}, \mathcal{G})$ ◁ Fits structural equation $\hat{f}_i$ for each x_i on input Pa_{x_i} using L_2 loss
for $\alpha \in 2^{\mathcal{R}_{\text{cand}}}$ **do**
  $\alpha_t \leftarrow \text{TopologicalSort}(\alpha, \mathcal{G})$ ◁ To apply the fixes in topological order
  Set $x_j^{\text{fix}} \sim P(X_j|\text{Pa}_{x_j}) \forall j \in \alpha_t$ ◁ Follow order α_t
  $\widehat{\boldsymbol{x}}^{\text{int}(\alpha)} \leftarrow \text{Intervene}(\widehat{\mathcal{S}}, \boldsymbol{x}, \text{fix})$ ◁ Intervene on α using the SCM $\widehat{\mathcal{S}}$
  $\phi[\alpha] \leftarrow g_n(x_n) - \mathbb{E}[g_n(\widehat{x}_n^{\text{int}(\alpha)})]$ ◁ Shapley utility for the subset α
end for
$\widehat{\mathcal{R}} \leftarrow \text{ShapleySort}(\phi, \text{ desc})$ ◁ Compute Shapley values and Sort $\widehat{\mathcal{R}}$ descending
return $\widehat{\mathcal{R}}$

B PROOFS

Definition 8 ((Redko et al., 2019)) *For a loss function ℓ and hypothesis class $\mathcal{H}$, we define the discrepancy distance between two distributions P, Q as follows:*

$$disc_\ell^{\mathcal{H}}(P, Q) = \sup_{h, h' \in \mathcal{H}^2} |\mathbb{E}_{x\sim P}[\ell(h(x), h'(x))] - \mathbb{E}_{x\sim Q}[\ell(h(x), h'(x))]|$$

Lemma 9 *For bounded loss functions $\ell(\bullet, \bullet) \leq M$, where $M > 0$, we have:*

$$disc_\ell^{\mathcal{H}}(P, Q) \leq M tvd(P, Q)$$

See (Redko et al., 2019) (Proposition 3.1) for the proof.

Theorem 10 *Suppose the true SCM $\mathcal{S}$ is an additive noise model defined over a chain graph $\mathcal{G} = X_1 \to \ldots \to X_n$ with structural equations of the form $f_i(x_{i-1}) + \epsilon_i$, where ϵ_i has bounded variance σ^2 and f_i is K-Lipschitz. Let $\mathcal{H} = \{\mathcal{H}_i\}_{i=1}^n$ be the realizable hypothesis class for $\mathcal{S}$, with each $\mathcal{H}_i$ containing bounded functions that are K-lipschitz. Let $\widehat{\mathcal{S}}$ denote the SCM learned from training data D_{trn}. $\widehat{\mathcal{S}}$ encompasses the estimated functions $\{\hat{f}_i\}_{i=1}^n$. For any j, let $Q_X^{RC(j)}$ denote the distribution of samples with a unique root cause at X_j. Then, for $\boldsymbol{x}$ sampled from $Q_X^{RC(j)}$ with fix $x_j^{fix} \sim P_X^{\hat{trn}}(X_j|X_{j-1} = x_{j-1})$ applied to the root cause X_j, the error in estimated counterfactual at the target node X_n admits the following bound:*

$$\mathbb{E}_{\boldsymbol{x}\sim Q_X^{RC(j)}}[|x_n^{CF(j)} - \widehat{x}_n^{CF(j)}|] \leq \sum_{i>j} K^{n-i}\Bigg[2^{n-i+1}\mathbb{E}_{x_{i-1}\sim P_{X_{i-1}}^{trn}}\left[|f_i(x_{i-1}) - \hat{f}_i(x_{i-1})|\right] + M^{n-i+1}\cdot\left[tvd\left(P_{X_{i-1}}^{trn}, Q_{X_{i-1}}^{RC(j)}\right) + \left(tvd\left(P_{X_{i-1}}^{trn}, P_{X_{i-1}}^{\hat{trn}}\right)\right)\right]\Bigg]$$

Proof For the intervention x_j^{fix} applied on X_j, let us denote the true counterfactual values obtained from the SCM $\mathcal{S}$ as $\boldsymbol{x}^{\text{CF}(j)}$, and the estimated counterfactual from the learned SCM $\widehat{\mathcal{S}}$ as $\widehat{\boldsymbol{x}}^{\text{CF}(j)}$. Then, at $j+1$, we have:

$$x_{j+1}^{\text{CF}(j)} = f_{j+1}(x_j^{\text{fix}}) + \epsilon_{j+1} \text{ where } \epsilon_{j+1} = x_{j+1} - f_{j+1}(x_j) \tag{8}$$

$$\widehat{x}_{j+1}^{\text{CF}(j)} = \hat{f}_{j+1}(x_j^{\text{fix}}) + \hat{\epsilon}_{j+1} \text{ where } \hat{\epsilon}_{j+1} = x_{j+1} - \hat{f}_{j+1}(x_j) \tag{9}$$

Now, let us bound the error in the true CF $\boldsymbol{x}^{\text{CF}(j)}$ and the estimated CF $\widehat{\boldsymbol{x}}^{\text{CF}(j)}$ using $\widehat{\mathcal{S}}$ at index $j+1$. Taking difference of Eqs. 8, 9

$$|x_{j+1}^{\text{CF}(j)} - \widehat{x}_{j+1}^{\text{CF}(j)}| = |f_{j+1}(x_j^{\text{fix}}) - f_{j+1}(x_j) - \hat{f}_{j+1}(x_j^{\text{fix}}) + \hat{f}_{j+1}(x_j)| \tag{10}$$

$$\leq |f_{j+1}(x_j^{\text{fix}}) - \hat{f}_{j+1}(x_j^{\text{fix}})| + |f_{j+1}(x_j) - \hat{f}_{j+1}(x_j)| \tag{11}$$

$$\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|x_{j+1}^{\text{CF}(j)} - \widehat{x}_{j+1}^{\text{CF}(j)}|] \leq \underbrace{\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}\left[\mathbb{E}_{x_j^{\text{fix}}\sim P_{X_j}^{\hat{\text{trn}}}(X_j|x_{j-1})}\left[|f_{j+1}(x_j^{\text{fix}}) - \hat{f}_{j+1}(x_j^{\text{fix}})|\right]\right]}_{\text{term 1}}$$

$$+ \underbrace{\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|f_{j+1}(x_j) - \hat{f}_{j+1}(x_j)|]}_{\text{term 2}} \tag{12}$$

Let us bound term 2 above.

Recall that $\widehat{\mathcal{S}}$ is obtained by training from D_{trn}, $\hat{f}_{j+1}$ is learned using N samples obtained i.i.d. from $P_{X_j}^{\text{trn}}$. Further, the loss is assessed only for X_{j+1}; we have $\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|f_{j+1}(x_j) - \hat{f}_{j+1}(x_j)|] = \mathbb{E}_{x_j\sim Q_{X_j}^{\text{RC}(j)}}[|f_{j+1}(x_j) - \hat{f}_{j+1}(x_j)|]$.

Therefore, we have:

$$\mathbb{E}_{x_j\sim Q_{X_j}^{\text{RC}(j)}}[|f_{j+1}(x_j) - \hat{f}_{j+1}(x_j)|] \tag{13}$$

$$= \mathbb{E}_{x_j\sim Q_{X_j}^{\text{RC}(j)}}[|f_{j+1}(x_j) - \hat{f}_{j+1}(x_j)|] + \mathbb{E}_{x_j\sim P_{X_j}^{\text{trn}}}[|f_{j+1}(x_j) - \hat{f}_{j+1}(x_j)|]$$
$$- \mathbb{E}_{x_j\sim P_{X_j}^{\text{trn}}}[|f_{j+1}(x_j) - \hat{f}_{j+1}(x_j)|]$$

$$\leq \mathbb{E}_{x_j\sim P_{X_j}^{\text{trn}}}[|f_{j+1}(x_j) - \hat{f}_{j+1}(x_j)|] + \int_{x_j} |f_{j+1}(x_j) - \hat{f}_{j+1}(x_j)| \cdot |P_{X_j}^{\text{trn}}(x_j) - Q_{X_j}^{\text{RC}(j)}(x_j)| dx_j \tag{14}$$

$$\leq \mathbb{E}_{x_j\sim P_{X_j}^{\text{trn}}}[|f_{j+1}(x_j) - \hat{f}_{j+1}(x_j)|] \tag{15}$$
$$+ \sup_{h,h'\in\mathcal{H}_{j+1}^2} \int_{x_j} |h(x_j) - h'(x_j)| \cdot |P_{X_j}^{\text{trn}}(x_j) - Q_{X_j}^{\text{RC}(j)}(x_j)| dx_j$$

$$= \mathbb{E}_{x_j\sim P_{X_j}^{\text{trn}}}[|f_{j+1}(x_j) - \hat{f}_{j+1}(x_j)|] + \text{disc}_{\ell_1}^{\mathcal{H}_{j+1}}(P_{X_j}^{\text{trn}}(x_j), Q_{X_j}^{\text{RC}(j)}(x_j)) \tag{16}$$

The last inequality is valid as long as $f_{j+1} \in \mathcal{H}_{j+1}$ which is true because $\mathcal{H}_{j+1}$ is realizable. Further since $\mathcal{H}_{j+1}$ encompasses bounded functions, we can bound the L_1 loss ℓ_1 using a constant $M > 0$. Therefore,

$$\mathbb{E}_{\boldsymbol{x}\sim Q_{X_j}^{\text{RC}(j)}}[|f_{j+1}(x_j) - \hat{f}_{j+1}(x_j)|] \leq \underbrace{\mathbb{E}_{x_j\sim P_{X_j}^{\text{trn}}}[|f_{j+1}(x_j) - \hat{f}_{j+1}(x_j)|]}_{\text{term 1}} + \underbrace{M\text{tvd}(P_{X_j}^{\text{trn}}(x_j), Q_{X_j}^{\text{RC}(j)}(x_j))}_{\text{term 2}} \tag{17}$$

where term 1 represents a classical divergence between empirical risk and true risk, and can be bounded using classical VC, Rademacher complexity based generalization bounds. Term 2 is more interesting as it captures the divergence between the standard training distribution and the root cause distribution that governs when an anomaly occurs.

Combining Eqs. 12, 17 for the error at X_{j+1}, we get

$$\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|x_{j+1}^{\text{CF}(j)} - \widehat{x}_{j+1}^{\text{CF}(j)}|] \leq \underbrace{\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}\left[\mathbb{E}_{x_j^{\text{fix}}\sim P_{X_j}^{\hat{\text{trn}}}(X_j|x_{j-1})}\left[|f_{j+1}(x_j^{\text{fix}}) - \hat{f}_{j+1}(x_j^{\text{fix}})|\right]\right]}_{\text{term 1}} \tag{18}$$

$$+ \underbrace{\mathbb{E}_{x_j\sim P_{X_j}^{\text{trn}}}[|f_{j+1}(x_j) - \hat{f}_{j+1}(x_j)|] + M \cdot \text{tvd}(P_{X_j}^{\text{trn}}, Q_{X_j}^{\text{RC}(j)})}_{\text{term 2}}$$

Now, we reduce the first term above. For a fix $x_j^{\text{fix}} \sim P^{\hat{\text{trn}}}_{X_j}(X_j|x_{j-1})$, the following holds:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}\left[\mathbb{E}_{x_j^{\text{fix}}\sim P^{\hat{\text{trn}}}_{X_j}(X_j|x_{j-1})}\left[|f_{j+1}(x_j^{\text{fix}}) - \hat{f}_{j+1}(x_j^{\text{fix}})|\right]\right] && (19)\\
&= \mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}\left[\mathbb{E}_{x_j^{\text{fix}}\sim P^{\hat{\text{trn}}}_{X_j}(X_j|x_{j-1})}\left[|f_{j+1}(x_j^{\text{fix}}) - \hat{f}_{j+1}(x_j^{\text{fix}})|\right]\right]\\
&+ \mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}\left[\mathbb{E}_{x_j^{\text{fix}}\sim P^{\text{trn}}_{X_j}(X_j|x_{j-1})}\left[|f_{j+1}(x_j^{\text{fix}}) - \hat{f}_{j+1}(x_j^{\text{fix}})|\right]\right]\\
&- \mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}\left[\mathbb{E}_{x_j^{\text{fix}}\sim P^{\text{trn}}_{X_j}(X_j|x_{j-1})}\left[|f_{j+1}(x_j^{\text{fix}}) - \hat{f}_{j+1}(x_j^{\text{fix}})|\right]\right]\\
&\leq \mathbb{E}_{x_j\sim P^{\text{trn}}_{X_j}}\left[|f_{j+1}(x_j) - \hat{f}_{j+1}(x_j)|\right] + M\cdot \text{tvd}(P^{\text{trn}}_{X_j}, P^{\hat{\text{trn}}}_{X_j}) && (20)
\end{aligned}$$

This follows from the definition of $Q^{\text{RC}(j)}$ 1. First observe that for the indices till $\{1,\ldots,j-1\}$, both root cause distribution $Q_X^{\text{RC}(j)}$ and the training distribution P_X^{trn} agree; i.e., $Q_X^{\text{RC}(j)}(X_1,\ldots,X_{j-1}) = P_X^{\text{trn}}(X_1,\ldots,X_{j-1})$. The only index that changes is j at which the root cause occurs. But since, while applying a fix, we sample it from $P_X^{\text{trn}}(X_j|x_{j-1})$, the marginal distribution of the fix simply reduces to $P^{\text{trn}}_{X_j}$.

Finally, combining the inequalities in 18, 20, we get:

$$\boxed{\begin{aligned}
\mathbb{E}_{x_j\sim Q_X^{\text{RC}(j)}}[|x_{j+1}^{\text{CF}(j)} - \widehat{x}_{j+1}^{\text{CF}(j)}|] &\leq 2\mathbb{E}_{x_j\sim P^{\text{trn}}_{X_j}}[|f_{j+1}(x_j) - \hat{f}_{j+1}(x_j)|] \\
&+ M\cdot\left[\text{tvd}(P^{\text{trn}}_{X_j}, Q^{\text{RC}(j)}_{X_j}) + \text{tvd}(P^{\text{trn}}_{X_j}, P^{\hat{\text{trn}}}_{X_j})\right]
\end{aligned} \quad (21)}$$

Now, let us carry forward these arguments to $j+2$.

$$x_{j+2}^{\text{CF}(j)} = f_{j+2}(x_{j+1}^{\text{CF}(j)}) + \epsilon_{j+2} \text{ where } \epsilon_{j+2} = x_{j+2} - f_{j+2}(x_{j+1}) \quad (22)$$

$$\widehat{x}_{j+2}^{\text{CF}(j)} = \hat{f}_{j+2}(\widehat{x}_{j+1}^{\text{CF}(j)}) + \hat{\epsilon}_{j+2} \text{ where } \hat{\epsilon}_{j+2} = x_{j+2} - \hat{f}_{j+2}(x_{j+1}) \quad (23)$$

$$(24)$$

Now for the estimated counterfactual at X_{j+2}, we have the error

$$\begin{aligned}
|x_{j+2}^{\text{CF}(j)} - \widehat{x}_{j+2}^{\text{CF}(j)}| &= |f_{j+2}(x_{j+1}^{\text{CF}(j)}) - f_{j+2}(x_{j+1}) - \hat{f}_{j+2}(\widehat{x}_{j+1}^{\text{CF}(j)}) + \hat{f}_{j+2}(x_{j+1})| && (25)\\
&\leq \underbrace{|f_{j+2}(x_{j+1}^{\text{CF}(j)}) - \hat{f}_{j+2}(\widehat{x}_{j+1}^{\text{CF}(j)})|}_{\text{term 1}} + \underbrace{|f_{j+2}(x_{j+1}) - \hat{f}_{j+2}(x_{j+1})|}_{\text{term 2}} && (26)
\end{aligned}$$

The term 2 above admits the same proof technique as we derived for X_{j+1}.

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|f_{j+2}(x_{j+1}) - \hat{f}_{j+2}(x_{j+1})|] &\leq \mathbb{E}_{x_{j+1}\sim P^{\text{trn}}_{X_{j+1}}}[|f_{j+2}(x_{j+1}) - \hat{f}_{j+2}(x_{j+1})|] \quad (27)\\
&+ M\cdot \text{tvd}(P^{\text{trn}}_{X_{j+1}}, Q^{\text{RC}(j)}_{X_{j+1}})
\end{aligned}$$

Now, let us analyze the first term.

Since $\hat{f}_{j+2}$ is K-Lipschitz, we have $|\hat{f}_{j+2}(\widehat{x}_{j+1}^{\text{CF}(j)}) - \hat{f}_{j+2}(x_{j+1}^{\text{CF}(j)})| \leq K|\widehat{x}_{j+1}^{\text{CF}(j)} - x_{j+1}^{\text{CF}(j)}|$. Therefore,

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|f_{j+2}(x_{j+1}^{\text{CF}(j)}) - \hat{f}_{j+2}(\widehat{x}_{j+1}^{\text{CF}(j)})|] && (28)\\
&= \mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|f_{j+2}(x_{j+1}^{\text{CF}(j)}) - \hat{f}_{j+2}(x_{j+1}^{\text{CF}(j)}) + \hat{f}_{j+2}(x_{j+1}^{\text{CF}(j)}) - \hat{f}_{j+2}(\widehat{x}_{j+1}^{\text{CF}(j)})|]\\
&\leq \mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|f_{j+2}(x_{j+1}^{\text{CF}(j)}) - \hat{f}_{j+2}(x_{j+1}^{\text{CF}(j)})|] + \mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|\hat{f}_{j+2}(x_{j+1}^{\text{CF}(j)}) - \hat{f}_{j+2}(\widehat{x}_{j+1}^{\text{CF}(j)})|] && (29)\\
&\leq \mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|f_{j+2}(x_{j+1}^{\text{CF}(j)}) - \hat{f}_{j+2}(x_{j+1}^{\text{CF}(j)})|] + K\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|x_{j+1}^{\text{CF}(j)} - \widehat{x}_{j+1}^{\text{CF}(j)}|] && (30)
\end{aligned}$$

Using the same arguments as used in Eq. 20, it can be shown that:

$$\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|f_{j+2}(x_{j+1}^{\text{CF}(j)}) - \hat{f}_{j+2}(x_{j+1}^{\text{CF}(j)})|] \leq \mathbb{E}_{x_{j+1}\sim P_{X_{j+1}}^{\text{trn}}}[|f_{j+2}(x_{j+1}) - \hat{f}_{j+2}(x_{j+1})|] + M \cdot \text{tvd}(P_{X_{j+1}}^{\text{trn}}, P_{X_{j+1}}^{\widehat{\text{trn}}}) \tag{31}$$

Finally, we have:

$$\boxed{\begin{aligned}\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|x_{j+2}^{\text{CF}(j)} - \widehat{x}_{j+2}^{\text{CF}(j)}|] \leq\ & 2\mathbb{E}_{x_{j+1}\sim P_{X_{j+1}}^{\text{trn}}}[|f_{j+2}(x_{j+1}) - \hat{f}_{j+2}(x_{j+1})|] \\ & + K\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|x_{j+1}^{\text{CF}(j)} - \widehat{x}_{j+1}^{\text{CF}(j)}|] \\ & + M \cdot \left[\text{tvd}(P_{X_{j+1}}^{\text{trn}}, Q_{X_{j+1}}^{\text{RC}(j)}) + \text{tvd}(P_{X_{j+1}}^{\text{trn}}, P_{X_{j+1}}^{\widehat{\text{trn}}})\right]\end{aligned}} \tag{32}$$

Now, we can extend this result to assess the error at X_n as follows:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|x_n^{\text{CF}(j)} - \widehat{x}_n^{\text{CF}(j)}|] \leq \sum_{i>j} K^{n-i}\Bigg[& 2^{n-i+1}\mathbb{E}_{x_{i-1}\sim P_{X_{i-1}}^{\text{trn}}}\left[|f_i(x_{i-1}) - \hat{f}_i(x_{i-1})|\right] \\ & + M^{n-i+1} \cdot \left[\text{tvd}\left(P_{X_{i-1}}^{\text{trn}}, Q_{X_{i-1}}^{\text{RC}(j)}\right) + \left(\text{tvd}\left(P_{X_{i-1}}^{\text{trn}}, P_{X_{i-1}}^{\widehat{\text{trn}}}\right)\right)\right]\Bigg]\end{aligned}$$

The above inequality holds because from definition of $Q_X^{\text{RC}(j)}$ 1, we have $\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|x_i^{\text{CF}(j)} - \widehat{x}_i^{\text{CF}(j)}|] = 0$ for any $i < j$. Further at X_j, because of performing an intervention we have $x_j^{\text{CF}(j)} = \widehat{x}_j^{\text{CF}(j)} = x_j^{\text{fix}}$.

■

Remark: Several prior works (Chen et al., 2014; Lin et al., 2018; Liu et al., 2021) defined root cause as a node that is anomalous and is connected to target node X_n through a chain of anomalous nodes. i.e., they expect $\varphi_i(x_i) = 1$ for all $i \geq j$, and in such a case, we see that tvd $\left(P_{X_i}^{\text{trn}}, Q_{X_i}^{\text{RC}(j)}\right)$ is large for any $i > j$. Nonetheless, the leading term in the above geometric progression has tvd $\left(P_{X_j}^{\text{trn}}, Q_{X_k}^{\text{RC}(j)}\right)$ for which we know $\varphi_j(x_j) = 1$ by root cause definition, and therefore, in practice $\widehat{\boldsymbol{x}}^{\text{CF}(j)}$ is a poor estimate for $\boldsymbol{x}^{\text{CF}(j)}$. All the other terms can be bounded using classical generalization bounds, and they go down with the size of training data $|D_{\text{trn}}|$.

Theorem 11 *Under the same conditions laid out in Theorem 5, the error between the true counterfactual $\boldsymbol{x}^{CF(j)}$ and the estimated intervention $\widehat{\boldsymbol{x}}^{int(j)}$ admits the following bound:*

$$\begin{aligned}\mathbb{E}_{\boldsymbol{x}\sim Q_X^{RC(j)}}\left[|x_n^{CF(j)} - \widehat{x}_n^{int(j)}|\right] \leq \sum_{i>j} K^{n-i}\Bigg[& \mathbb{E}_{x_{i-1}\sim P_{X_{i-1}}^{trn}}\left[|f_i(x_{i-1}^{CF(j)}) - \hat{f}_i(x_{i-1}^{CF(j)})|\right] \\ & + M^{n-i+1} \cdot \left(tvd\left(P_{X_{i-1}}^{trn}, P_{X_{i-1}}^{\widehat{trn}}\right)\right) + std(\epsilon_i) + std(\tilde{\epsilon}_i) + |\mathbb{E}[\epsilon_i] - \mathbb{E}[\tilde{\epsilon}_i]|\Bigg]\end{aligned}$$

where bias$(\hat{f}_i) = \mathbb{E}_{x_i\sim P_{X_i}^{trn}}[f_i(x_i) - \hat{f}_i(x_i)]$. For an unbiased learner, bias is zero.

Proof For the intervention x_j^{fix} applied on X_j, let us denote the true counterfactual values obtained from the SCM $\mathcal{S}$ as $\boldsymbol{x}^{\text{CF}(j)}$, and the estimated intervention from the learned SCM $\widehat{\mathcal{S}}$ as $\widehat{\boldsymbol{x}}^{\text{int}(j)}$. Then, at $j+1$, we have:

$$x_{j+1}^{\text{CF}(j)} = f_{j+1}(x_j^{\text{fix}}) + \epsilon_{j+1} \text{ where } \epsilon_{j+1} = x_{j+1} - f_{j+1}(x_j) \tag{33}$$

$$\widehat{x}_{j+1}^{\text{int}(j)} = \hat{f}_{j+1}(x_j^{\text{fix}}) + \tilde{\epsilon}_{j+1} \text{ where } \tilde{\epsilon}_{j+1} \sim P_{\epsilon_{j+1}}^{\widehat{\text{trn}}} \tag{34}$$

where, $P^{\hat{\text{trn}}}_{\epsilon_{j+1}}$ represents the empirical distribution of the marginal $P^{\text{trn}}_{\epsilon_{j+1}}$, obtained from a validation dataset $D_V \subset D_{\text{trn}}$. For additive noise models, this is simply the empirical distribution of the error residuals, $x_{j+1} - \hat{f}_{j+1}(x_j)$ for $\boldsymbol{x} \in D_V$.

Now, let us bound the error in the true CF $\boldsymbol{x}^{\text{CF}(j)}$ and the estimated intervention $\widehat{\boldsymbol{x}}^{\text{int}(j)}$ using $\widehat{\mathcal{S}}$ at index $j+1$. Taking difference of above Eqs.

$$\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|x_{j+1}^{\text{CF}(j)} - \widehat{x}_{j+1}^{\text{int}(j)}|] = \underbrace{\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}\left[\mathbb{E}_{x_j^{\text{fix}}\sim P^{\hat{\text{trn}}}_{X_j}(X_j|x_{j-1})}\left[|f_{j+1}(x_j^{\text{fix}}) - \hat{f}_{j+1}(x_j^{\text{fix}})|\right]\right]}_{\text{term 1}} + \underbrace{\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}\mathbb{E}_{\tilde{\epsilon}_{j+1}\sim\hat{\mathcal{P}}_{\epsilon_{j+1}}}[|\epsilon_{j+1} - \tilde{\epsilon}_{j+1}|]}_{\text{term 2}} \tag{35}$$

We use Eq. 20 to reduce term 1 to $\mathbb{E}_{x_j\sim P^{\text{trn}}_{X_j}}\left[|f_{j+1}(x_j) - \hat{f}_{j+1}(x_j)|\right] + M \cdot \text{tvd}(P^{\text{trn}}_{X_j}, P^{\hat{\text{trn}}}_{X_j})$.

Now, let us analyze term 2. Since X_j is the unique root cause, we have $Q^{\text{RC}(j)}_{\epsilon_{j+1}} = P^{\text{trn}}_{\epsilon_{j+1}}$. Therefore,

$$\mathbb{E}_{\epsilon_{j+1}\sim Q^{\text{RC}(j)}_{\epsilon_{j+1}}}\mathbb{E}_{\tilde{\epsilon}_{j+1}\sim\hat{\mathcal{P}}_{\epsilon_{j+1}}}[|\epsilon_{j+1} - \mathbb{E}[\tilde{\epsilon}_{j+1}]|] = \mathbb{E}_{\epsilon_{j+1}\sim P^{\text{trn}}_{\epsilon_{j+1}}}\mathbb{E}_{\tilde{\epsilon}_{j+1}\sim\hat{\mathcal{P}}_{\epsilon_{j+1}}}[|\epsilon_{j+1} - \mathbb{E}[\tilde{\epsilon}_{j+1}]|] \tag{36}$$

Now,

$$|\epsilon_{j+1} - \tilde{\epsilon}_{j+1}| = |\epsilon_{j+1} - \mathbb{E}[\epsilon_{j+1}] + \mathbb{E}[\epsilon_{j+1}] - \mathbb{E}[\tilde{\epsilon}_{j+1}] + \mathbb{E}[\tilde{\epsilon}_{j+1}] - \tilde{\epsilon}_{j+1}| \tag{37}$$

$$\leq |\epsilon_{j+1} - \mathbb{E}[\epsilon_{j+1}]| + |\mathbb{E}[\epsilon_{j+1}] - \mathbb{E}[\tilde{\epsilon}_{j+1}]| + |\mathbb{E}[\tilde{\epsilon}_{j+1}] - \tilde{\epsilon}_{j+1}| \tag{38}$$

using $|a+b| \leq |a| + |b|$. Note that $|\mathbb{E}[\epsilon_{j+1}] - \mathbb{E}[\tilde{\epsilon}_{j+1}]|$ is a constant.

Taking expectation on both sides, we get:

$$\mathbb{E}_{\epsilon_{j+1}\sim P^{\text{trn}}_{\epsilon_{j+1}}}\mathbb{E}_{\tilde{\epsilon}_{j+1}\sim\hat{\mathcal{P}}_{\epsilon_{j+1}}}[|\epsilon_{j+1} - \mathbb{E}[\tilde{\epsilon}_{j+1}]|] \tag{39}$$

$$\leq \mathbb{E}_{\epsilon_{j+1}\sim P^{\text{trn}}_{\epsilon_{j+1}}}[|\epsilon_{j+1} - \mathbb{E}[\epsilon_{j+1}]|] + \mathbb{E}_{\tilde{\epsilon}_{j+1}\sim\hat{\mathcal{P}}_{\epsilon_{j+1}}}[|\mathbb{E}[\tilde{\epsilon}_{j+1}] - \tilde{\epsilon}_{j+1}|] + |\mathbb{E}[\epsilon_{j+1}] - \mathbb{E}[\tilde{\epsilon}_{j+1}]|$$

$$\leq \sqrt{\mathbb{E}_{\epsilon_{j+1}\sim P^{\text{trn}}_{\epsilon_{j+1}}}[(\epsilon_{j+1} - \mathbb{E}[\epsilon_{j+1}])^2]} + \sqrt{\mathbb{E}_{\tilde{\epsilon}_{j+1}\sim\hat{\mathcal{P}}_{\epsilon_{j+1}}}[(\mathbb{E}[\tilde{\epsilon}_{j+1}] - \tilde{\epsilon}_{j+1})^2]} + |\mathbb{E}[\epsilon_{j+1}] - \mathbb{E}[\tilde{\epsilon}_{j+1}]| \tag{40}$$

$$= \text{std}(\epsilon_{j+1}) + \text{std}(\tilde{\epsilon}_{j+1}) + |\mathbb{E}[\epsilon_{j+1}] - \mathbb{E}[\tilde{\epsilon}_{j+1}]| \tag{41}$$

where $\text{std}(\epsilon_{j+1})$ is the standard deviation of the latent noise ϵ_{j+1}.

Remark: It is common in causal literature to assume that ϵ_i are zero-mean in addition to having bounded variance. When zero mean assumption holds, we can use $\tilde{\epsilon}_{j+1} = 0$, and in that case, we will have $\mathbb{E}_{\epsilon\sim Q^{\text{RC}(j)}_{\epsilon_{j+1}}}\mathbb{E}_{\tilde{\epsilon}_{j+1}\sim\hat{\mathcal{P}}_{\epsilon_{j+1}}}[|\epsilon_{j+1} - \mathbb{E}[\tilde{\epsilon}_{j+1}]|] \leq \sigma(\epsilon_{j+1})$.

Finally, for the estimated intervention at X_j, we have:

$$\boxed{\begin{aligned}\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|x_{j+1}^{\text{CF}(j)} - \widehat{x}_{j+1}^{\text{int}(j)}|] &\leq \mathbb{E}_{x_j\sim P^{\text{trn}}_{X_j}}\left[|f_{j+1}(x_j) - \hat{f}_{j+1}(x_j)|\right] \\ &+ M \cdot \text{tvd}(P^{\text{trn}}_{X_j}, P^{\hat{\text{trn}}}_{X_j}) + \text{std}(\epsilon_{j+1}) + \text{std}(\tilde{\epsilon}_{j+1}) + |\mathbb{E}[\epsilon_{j+1}] - \mathbb{E}[\tilde{\epsilon}_{j+1}]|\end{aligned}} \tag{42}$$

For the estimated intervention at X_{j+2}, we have:

$$x_{j+2}^{\text{CF}(j)} = f_{j+2}(x_{j+1}^{\text{CF}(j)}) + \epsilon_{j+2} \text{ where } \epsilon_{j+2} = x_{j+2} - f_{j+2}(x_{j+1}) \tag{43}$$

$$\widehat{x}_{j+2}^{\text{int}(j)} = \hat{f}_{j+2}(\widehat{x}_{j+1}^{\text{int}(j)}) + \mathbb{E}[\tilde{\epsilon}_{j+2}] \text{ where } \tilde{\epsilon}_{j+2} \sim P^{\hat{\text{trn}}}_{\epsilon_{j+2}} \tag{44}$$

Therefore, the error at $j+2$ is:

$$\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|x_{j+2}^{\text{CF}(j)} - \widehat{x}_{j+2}^{\text{int}(j)}|] = \underbrace{\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}\left[|f_{j+2}(x_{j+1}^{\text{CF}(j)}) - \hat{f}_{j+2}(\widehat{x}_{j+1}^{\text{int}(j)})|\right]}_{\text{term 1}} + \underbrace{\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|\epsilon_{j+2} - \mathbb{E}[\tilde{\epsilon}_{j+2}]|]}_{\text{term 2}} \tag{45}$$

We know that we can bound $\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|\epsilon_{j+2} - \mathbb{E}[\tilde{\epsilon}_{j+2}]|] \leq \text{std}(\epsilon_{j+2}) + \text{std}(\tilde{\epsilon}_{j+2}) + |\mathbb{E}[\epsilon_{j+2}] - \mathbb{E}[\tilde{\epsilon}_{j+2}]|$.

To bound the first term, we use Lipschitz property.

$$\begin{aligned}\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|f_{j+2}(x_{j+1}^{\text{CF}(j)}) - \hat{f}_{j+2}(x_{j+1}^{\text{CF}(j)})|] &\leq \mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|f_{j+2}(x_{j+1}^{\text{CF}(j)}) - \hat{f}_{j+2}(x_{j+1}^{\text{CF}(j)})|] \qquad (46)\\ &\quad + K\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|x_{j+1}^{\text{CF}(j)} - \widehat{x}_{j+1}^{\text{int}(j)}|]\\ &\leq \mathbb{E}_{x_{j+1}\sim P_{X_{j+1}}^{\text{trn}}}\left[|f_{j+2}(x_{j+1}) - \hat{f}_{j+2}(x_{j+1})|\right] \qquad (47)\\ &\quad + M\cdot \text{tvd}(P_{X_{j+1}}^{\text{trn}}, P_{X_{j+1}}^{\hat{\text{trn}}})\end{aligned}$$

Finally, we have:

$$\boxed{\begin{aligned}\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|x_{j+2}^{\text{CF}(j)} - \widehat{x}_{j+2}^{\text{int}(j)}|] &\leq \mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|f_{j+2}(x_{j+1}^{\text{CF}(j)}) - \hat{f}_{j+2}(x_{j+1}^{\text{CF}(j)})|] \qquad (48)\\ &\quad + K\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}[|x_{j+1}^{\text{CF}(j)} - \widehat{x}_{j+1}^{\text{int}(j)}|]\\ &\quad + \text{std}(\epsilon_{j+2}) + \text{std}(\tilde{\epsilon}_{j+2}) + |\mathbb{E}[\epsilon_{j+2}] - \mathbb{E}[\tilde{\epsilon}_{j+2}]|\end{aligned}}$$

We can extend this result ts assess the error at X_n as follows:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{x}\sim Q_X^{\text{RC}(j)}}\left[|x_n^{\text{CF}(j)} - \widehat{x}_n^{\text{int}(j)}|\right] \leq \sum_{i>j} K^{n-i}\Bigg[&\mathbb{E}_{x_{i-1}\sim P_{X_{i-1}}^{\text{trn}}}\left[|f_i(x_{i-1}^{\text{CF}(j)}) - \hat{f}_i(x_{i-1}^{\text{CF}(j)})|\right]\\ &+ M^{n-i+1}\cdot\left(\text{tvd}\left(P_{X_{i-1}}^{\text{trn}}, P_{X_{i-1}}^{\hat{\text{trn}}}\right)\right) + \text{std}(\epsilon_i) + \text{std}(\tilde{\epsilon}_i) + |\mathbb{E}[\epsilon_i] - \mathbb{E}[\tilde{\epsilon}_i]|\Bigg]\end{aligned}$$

■

Corollary 12 *Suppose in Theorem 6, the exogenous variables ϵ_i are zero-mean in addition to having bounded variance for any $i \in [n]$, then the error between interventions and counterfactuals admits the following bound:*

$$\begin{aligned}\mathbb{E}_{\boldsymbol{x}\sim Q_X^{RC(j)}}[|x_n^{CF(j)} - \widehat{x}_n^{int(j)}|] \leq \sum_{i>j} K^{n-i}\Bigg[&E_{x_{i-1}\sim P_{X_{i-1}}^{trn}}[|f_i(x_{i-1}^{CF(j)}) - \hat{f}_i(x_{i-1}^{CF(j)})|] \qquad (49)\\ &+ M^{n-i+1}\cdot\left(tvd\left(P_{X_{i-1}}^{trn}, P_{X_{i-1}}^{\hat{trn}}\right)\right) + std(\epsilon_i)\Bigg]\end{aligned}$$

Proof If we know that the latent exogenous variables are zero-mean, we can set $P_{\epsilon_{j+1}}^{\hat{\text{trn}}}$ in Eq. 34 to the dirac-delta distribution $\delta(\tilde{\epsilon}_{j+1} = 0)$.

Then we can see from the remark below Eq. 41 that the proof simply follows. ■

C Description of Datasets

C.1 Petshop

The PetShop application is a microservices-based pet adoption platform deployed on Amazon Web Services (AWS). It allows users to search for pets and complete adoption transactions. The system is built using multiple interconnected microservices, which include storage systems, publish-subscribe systems, load balancers, and other custom application based logic. These services were containerized and deployed using Kubernetes.

Causal Graph. The service dependencies in the PetShop application are captured in a directed graph, with edges indicating the call relationships between services. This call graph is inverted by reversing the edge directions to obtain the underlying causal graph for this dataset.

Node KPIs. The target node for root cause diagnosis is the PetSite node, the customer-facing frontend. When the website experiences abnormal latency in serving user requests, an anomaly is detected at PetSite, triggering the root cause diagnosis procedure. The target node has several ancestor nodes, including PetSearch ECS Fargate, petInfo DynamoDB Table, payforadoption ECS Fargate, lambdastatusupdater Lambda Function, and petlistadoptions ECS Fargate. Key Performance Indicators (KPIs) for these nodes were collected using Amazon CloudWatch, with system traces logged at 5-minute intervals. Metrics include the number of requests, and latency (both average and quantiles) for each microservice. The dataset spans three distinct traffic scenarios—low-traffic, high-traffic, and temporal-traffic—each defined by varying request patterns. For instance, the low-traffic scenario averages 484 requests per 5-minute window, while the high-traffic and temporal-traffic scenarios exhibit higher request rates. Therefore, diagnosing the root cause of High, and Temporal test cases is easier compared to the low latency ones. Overall, the dataset includes 68 injected issues across five nodes, with each issue associated with a unique ground-truth root cause. Consequently, each test case in this dataset features a **unique** root cause.

Root Cause Test cases. The issues injected into the system span various types, including request overloads, memory leaks, CPU hogs, misconfigurations, and artificial delays. These issues affect different services, such as petInfo DynamoDB Table, payforadoption ECS Fargate, and lambdastatusupdater Lambda Function. The delays are parameterized with varying severities and affect different services, such as:

- PetSearch ECS Fargate: 500-2000ms delays for bunny search requests
- payforadoption ECS Fargate: 250-1000ms delays for all requests
- Misconfiguration errors: Affecting 1-10% of requests depending on the service

C.2 Synthetic Datasets

We experiment with several synthetic datasets to benchmark the perfomance of the baselines against IDI. We desribe the generation of the causal graph first.

Causal Graph. We follow the graph generation procedure outlined in prior work (Budhathoki et al., 2022b). Given the total number of nodes n and the number of root nodes n_r, the causal graph is constructed as follows: The nodes $X_1, X_2, \ldots, X_{n_r}$ are assigned as root nodes. For each internal node X_k (where $k > n_r$), we randomly select its parent nodes from the set of preceding nodes $X_1, X_2, \ldots, X_{k-1}$. Each internal node has at most one parent, with a slight bias towards selecting a single parent. This sampling strategy helps to maintain a hierarchical structure and ensures the graph remains sufficiently deep. Once the parent nodes are chosen, directed edges are added from each parent to X_k. This process is repeated until all n nodes are connected, forming a directed acyclic graph. Finally, we randomly choose a node that is at least 10 levels deep from a root node to serve as the target anomalous node. We show some example causal graphs in Fig. 12.

Node KPIs. Since cloud KPIs take on positive values – such as latency, number of requests, throughput, and availability (Hardt et al., 2023; Meng et al., 2020)—which are the commonly logged metrics, we ensured that all the nodes remain positive. To achieve this at root nodes, we sample ϵ_i from a uniform distribution $U[0, 1]$. For the internal nodes, we generate their observed values using one of the following three approaches:

- **Linear:** For a node X_i with parents Pa_{X_i}, we define the functional equation linearly with random coefficients as: $x_i = w_i^\top \mathrm{Pa}_{x_i} + \epsilon_i$ where $w_i \sim U[0.5, 2]$. These coefficients ensure that the KPI remains positive. We did not use $U[0, 1]$ for the weights because doing so caused deeper nodes in the graph to diminish, eventually reaching zero.
- **Non-Linear Invertible:** For non-linearity, we define each local mechanism f_i as an additive noise three-layer ELU-activated MLP. We initialized the MLP weights using uniform distribution to ensure that even the internal nodes remain positive. Since many X_i nodes have only one parent, we found that using four hidden nodes in the MLP was sufficient to generate non-trivial test cases. In this case, the equation becomes: $X_i = \mathrm{mlp}(\mathrm{Pa}_{X_i}) + \epsilon_i$ where $\epsilon_i \sim U[0, 1]$.

- **Non-Linear Non-Invertible:** In this setting, we use the same MLP architecture as in the previous case, but without additive noise. Here, each MLP f_i receives both the parents Pa_{x_i} and the noise ϵ_i as inputs. Thus, the equation is: $X_i = \mathrm{mlp}(\mathrm{Pa}_{X_i}, \epsilon_i)$ with $\epsilon_i \sim U[0, 1]$. This configuration generates the most complex test cases in our experiments.

Root Cause Test cases. We consider two cases for the root cause test cases:

- **Unique:** In this case, we randomly sample a node from the ancestors of the target node.
- **Multiple:** Here, we randomly sample at most three nodes from the ancestors, ensuring that our assumption 1 holds. Specifically, no two root causes lie on the same path to the target. We experiment with test cases that satisfy this assumption and perform ablations to investigate its impact.

Suppose $\alpha^\star$ denotes the root cause node set. For $\alpha^\star$ to cause an anomaly at x_n, we apply a grid search over $\boldsymbol{\epsilon}_{\alpha^\star}$ such that ($\boldsymbol{\epsilon}_{\alpha^\star}$ they lead the SCM $\mathcal{S}$ to induce an anomaly at x_n. We start our search from 0 and increase them in steps of size 0.25 until the Z-score of the target node $\phi_n(x_n)$ hits the anomaly threshold 3.

D TIMING ANALYSIS

	Method	Petshop	Syn Linear	Syn Non-Linear
Correlation	Random Walk (Yu et al., 2021)	2.36	1.74	4.81
	Ranked Correlation (Hardt et al., 2023)	0.60	0.21	2.99
	ϵ-Diagnosis (Shan et al., 2019)	2.11	–	–
Causal Anomaly	Circa (Li et al., 2022)	0.52	0.36	2.73
	Traversal (Chen et al., 2014)	0.27	0.24	1.05
	Smooth Traversal (Okati et al., 2024)	0.30	0.26	0.99
Causal Fix	HRCD (Ikram et al., 2022)	11.69	–	–
	TOCA (Okati et al., 2024)	1.96	0.95	9.16
	CF Attribution (Budhathoki et al., 2022b)	9.71	22.99	178.47
Ours	IDI (CF)	0.42	0.38	8.31
	IDI	0.37	1.29	9.62

Table 5: Running time (in seconds) for datasets with **unique** root causes. "–" indicates that the baseline was not consider for the corresponding dataset.

We present the running time required for predicting the unique root cause across all methods for one test case in Table 5. We show the results for the semi-synthetic Petshop dataset, as well as the Linear and Non-Linear versions of our synthetic datasets. Note that we omit the Non-Linear Non-Invertible cases because their running times were comparable to the Non-Linear Invertible cases. We make the following observations:

1. The Correlation and Causal Anomaly methods demonstrate the best performance in terms of running time.
2. Causal Fix approaches, on the contrary, are bottlenecked by the need to learn the Structural Causal Model (SCM). Learning the SCM involves fitting a lightweight regression model $\hat{f}_i : \mathrm{Pa}_{X_i} \mapsto X_i$ for each node i. Recall that these models are lightweight because they only need to regress the parent covariates of the nodes. The Linear methods incur less time compared to the Non-Linear ones, as they can be learned using closed-form expressions, whereas Non-Linear methods require gradient descent-based training.
3. For predicting the unique root cause, both IDI (CF) and IDI do not require Shapley value computations, allowing them to run in significantly less time.
4. The baseline CF Attribution method, however, performs Shapley analysis across all nodes, even for the unique root cause, making it the worst-performing method in terms of running time.

Table 6 presents the results for the running time required to predict multiple root causes. Unlike the unique root cause, our method IDI and its CF ablation IDI (CF) require Shapley analysis in this setting. However, Shapley values are computed only for the subset of nodes in $\mathcal{R}_{\text{cand}}$, identified after the first step of our algorithm (the Anomaly condition). We make the following observations:

	Method	Syn Linear	Syn Non-Lin Inv	Syn Non-Lin Non-Inv
Corr.	Random Walk (Yu et al., 2021)	8.63	6.73	9.1
	Ranked Correlation (Hardt et al., 2023)	0.26	1.54	1.9
Causal Anomaly	Circa (Li et al., 2022)	0.38	2.34	2.23
	Traversal (Chen et al., 2014)	0.22	2.26	1.99
	Smooth Traversal (Okati et al., 2024)	0.26	2.37	1.95
Causal Fix	TOCA (Okati et al., 2024)	1.3	11.66	13.58
	CF Attribution (Budhathoki et al., 2022b)	23.48	120.61	190.08
Ours	IDI (CF)	7.08	40.12	73.41
	IDI	8.2	42.6	76.24

Table 6: Running time (in seconds) for datasets with **multiple** root causes.

1. The running time for the random walk-based approach increases due to the presence of multiple anomalous paths leading to the target node.
2. All other baseline approaches exhibit running times comparable to those observed in the unique root cause test cases.
3. The running times for IDI and IDI (CF) increase because of the additional Shapley computations. However, this increase is significantly smaller compared to CF Attribution, as the former computes Shapley values over a subset of nodes, while the latter evaluates them across all nodes.

Dataset	CF Attrib.	IDI (Ours)
Linear	36.4 ± 3.6	6.0 ± 1.6
Non-Linear Invertible	35.8 ± 4.2	6.0 ± 1.9
Non-Linear Non-Invertible	37.1 ± 3.8	6.2 ± 1.9

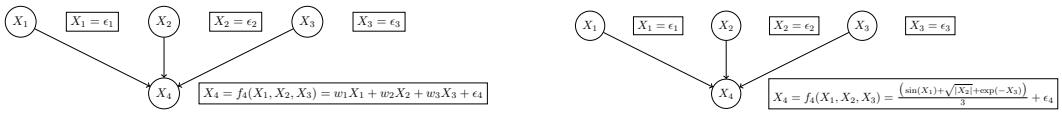
Table 7: Number of nodes considered for Shapley Analysis: We report the mean $\pm$ standard deviation computed across all the test cases.

We report the number of nodes involved in computing the Shapley values in Table 7. Since Shapley value computations are NP-Hard, in practice, Monte Carlo simulations are commonly used to approximate these values by sampling permutations of the nodes. In our work, we sampled 500 permutations for all methods to ensure tractability. Table 7 presents the mean number of nodes involved, along with the standard deviation across all test cases. Overall, we observe a $6\times$ reduction in the number of nodes for IDI compared to the CF Attribution baseline. Notably, if exact computation of Shapley values were performed, this reduction factor would be even more significant.

E Experiments with High Variance of ϵ_i

In this section, we experiment with different sampling distributions for ϵ_i to evaluate their impact on interventional and counterfactual estimates. We begin with a simple four-variable toy example.

E.1 Toy Setting



(a) Linear (b) Non-Linear

Figure 5: This figure illustrates a toy causal graph with four variables: root nodes X_1, X_2, and X_3, and a child node X_4, which is connected to all three root nodes. Panel (a) depicts a linear SCM where the root nodes are direct copies of the exogenous noise terms ϵ, while X_4 is a linear function of the root nodes. Panel (b) presents a non-linear version, where X_4 is the average of three non-linear functions applied to the root nodes. In both panels, the SCM follows an additive noise model.

As shown in Fig. 5, we consider two types of Oracle SCMs: a linear additive noise model (Fig. 5a) and a non-linear additive noise model (Fig. 5b). The linear weights w_1, w_2, w_3 are sampled from

the Gaussian distribution $\mathcal{N}(0,1)$. For non-linear case, we set X_4 as $\frac{\sin(X_1)+\sqrt{|X_2|}+\exp(-X_3)}{3}+\epsilon_4$. The exogenous variables of the root nodes $\epsilon_1, \epsilon_2, \epsilon_3$ are also sampled from $\mathcal{N}(0,1)$ during the usual regime. For the root cause test sample, we randomly set one of the nodes $\{X_1, X_2, X_3\}$ to a value sampled from $U[3,10]$. This makes sure that the root cause is abnormal and lies atleast 3 standard deviations away from its usual regime. The root cause node induces an abnormal behavior at the target node X_4.

	X_1	X_2	X_3	X_4
Training $\boldsymbol{x}^{\text{trn}}$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$f_4(\boldsymbol{x}^{\text{trn}}_{1:3})+\mathcal{N}(0,\sigma^2)$
Validation $\boldsymbol{x}^{\text{val}}$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$f_4(\boldsymbol{x}^{\text{val}}_{1:3})+\mathcal{N}(0,\sigma^2)$
Test $\boldsymbol{x}^{\text{RC}}$	$\mathcal{N}(0,1)$	$U[3,10]$	$\mathcal{N}(0,1)$	$f_4(\boldsymbol{x}^{\text{tst}}_{1:3})+\mathcal{N}(0,\sigma^2)$
Counterfactual $\widehat{\boldsymbol{x}}^{\text{CF}(2)}$	x_1^{RC}	$\mathcal{N}(0,1)$	x_3^{RC}	$f_4(\widehat{\boldsymbol{x}}^{\text{CF}(2)}_{1:3})+\hat{\epsilon}_4$
Intervention $\widehat{\boldsymbol{x}}^{\text{int}(2)}$	x_1^{RC}	$\mathcal{N}(0,1)$	x_3^{RC}	$f_4(\widehat{\boldsymbol{x}}^{\text{int}(2)}_{1:3})+\tilde{\epsilon}_4$

Table 8: This table outlines the process for generating each sample in our toy simulation study. The node X_4 draws its ϵ_4 from a normal distribution $\mathcal{N}(0,\sigma^2)$, with σ^2 varied between 0.25 and 25 in our experiments. For the test sample shown here, the root cause lies at X_2, and the corresponding fix is sampled from uniform $U[3,10]$. For counterfactuals, $\hat{\epsilon}_4$ represents the abducted ϵ, while for interventions, $\tilde{\epsilon}_4$ corresponds to a randomly sampled error residual obtained from the validation data.

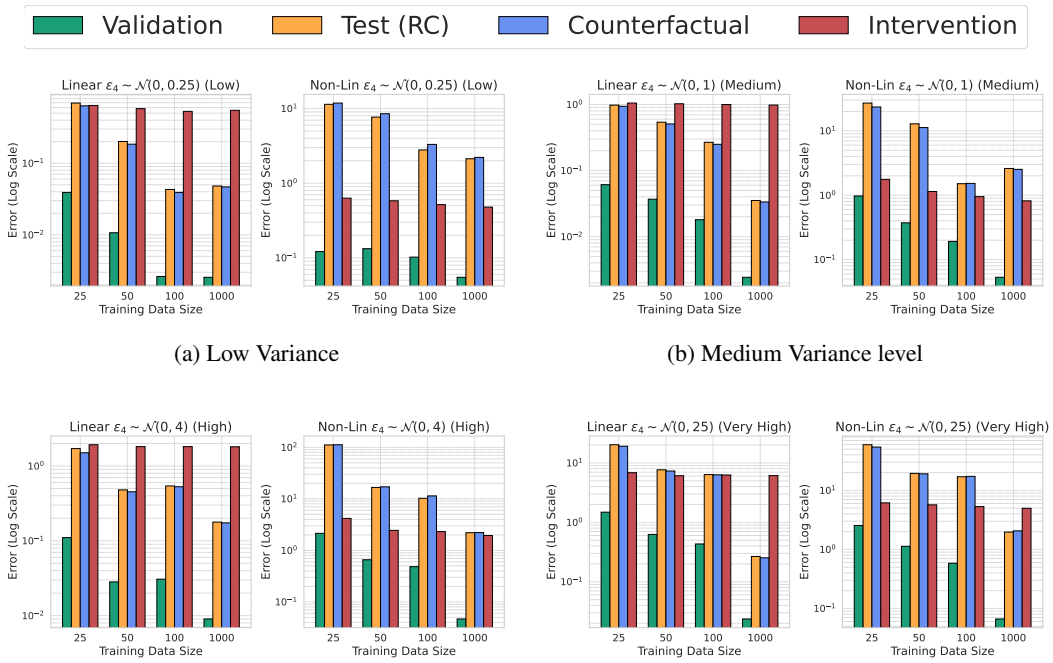
We generate n training samples where $n \in \{25, 50, 100, 1000\}$, and 100 validation samples. The test set also includes 100 samples, each with a unique root cause. For simplicity, we fix the abnormal parent X_j in the test instance by sampling x_j^{fix} from the ideal $\mathcal{N}(0,1)$ distribution. Table 8 summarizes the sampling procedure.

For this toy experiment, learning the SCM $\widehat{\mathcal{S}}$ amounts to fitting just one function $\hat{f}_4$. For the linear case, we fit a linear regression model with weights $\hat{w}_1, \hat{w}_2, \hat{w}_3$ and a bias $\hat{b}$, computed using a closed-form solution. For the non-linear case, we train a three-layer multi-layer perceptron with 10 hidden nodes and ReLU activations using gradient descent. We evaluate the following L_2 error metrics to analyze the impact of ϵ_4 variance on interventional vs. counterfactual estimates:

1. Validation Error: Measures the prediction accuracy of $\hat{f}_4$ on in-distribution data; $|f_4(\boldsymbol{x}^{\text{val}}_{1:3}) - \hat{f}_4(\boldsymbol{x}^{\text{val}}_{1:3})|^2$.
2. Test (RC) Error: Measures the prediction accuracy on OOD samples; $|f_4(\boldsymbol{x}^{\text{RC}}_{1:3}) - \hat{f}_4(\boldsymbol{x}^{\text{RC}}_{1:3})|^2$
3. Counterfactual Error: Quantifies the error arising from using CFs; $|x_4^{\text{CF}(j)} - \widehat{x}_4^{\text{CF}(j)}|^2$.
4. Interventional Error: Measures the error for using interventions; $|x_4^{\text{CF}(j)} - \widehat{x}_4^{\text{int}(j)}|^2$.

Figure 5 presents results for various settings of ϵ_4 variance. Each panel shows the linear setting on the left and the non-linear setting on the right. Key observations include:

- **Low Variance:** (a) A linear model, being convex, generalizes well across both training and test domains. As a result, the abducted values of $\hat{\epsilon}_4$ closely approximate the true values. This trend is evident in Fig. 6a, where the interventional error plateaus around the standard deviation 0.5, while the counterfactual (CF) error decreases with more training. (b) In contrast, for the non-linear model, interventional estimates significantly outperform CF estimates (note the $\log$ scale on the Y-axis). Nonlinear models are non-convex, leading to local minima or overfitting during training. Consequently, while validation error decreases rapidly, the test (RC) error remains high as training increases. This results in poor CF accuracy when $\hat{f}_4$ is evaluated on out-of-distribution (OOD) inputs.
- **Medium Variance:** Similar trends are observed in this setting. Figs. 6a and 6b together show that CF error correlates strongly with OOD test (RC) error, while interventional estimates remain bounded by the standard deviation. These results confirm the tightness of our theoretical bounds in practice.
- **High Variance:** In the linear dataset, the trends remain consistent. However, in the nonlinear dataset, interventional estimates show more pronounced advantages over CFs, especially in low-data regimes. This is because, in such scenarios, high variance ϵ_4 destabilizes $\hat{f}_4$'s training, leading to high variance in its parameters. Only with sufficient training samples (e.g., 1000) does



(a) Low Variance

(b) Medium Variance level

(c) High Variance

(d) Very High Variance

Figure 6: Assessing the impact of variance of ϵ_i using a four variable toy dataset.

the validation error approach zero, causing the CF error to be comparable with the interventional error.

- **Very High Variance:** In this regime, we observe the first instance where CF error outperforms interventional error for the nonlinear model. One interpretation of the extreme variance in ϵ_4 is that the features X_1, X_2, X_3 collected in the training data are insufficiently rich to explain the variance of X_4. This means that certain additional features must be collected, and thereby reduce the variance σ^2. Moreover, CFs perform better in this scenario only because their assumption of additive noise holds. We will demonstrate in the subsequent experiment that CFs perform very poorly when this assumption is violated.
- **Summary:** While interventions may appear less favorable for linear models, we observed them to be adequate for in the root cause diagnosis problem. This is because, even if the estimated $\widehat{x}_4^{\text{int}(2)}$ deviates from the true $x_4^{\text{CF}(j)}$, it suffices to infer the correct signal $\phi_4(x_4^{\text{CF}(2)})$ regarding whether the fix suppresses the root cause at the target. Consequently, both interventions and CFs achieve near-perfect recall in our linear experiments, as reported in the main paper. However, for non-linear models, interventions surely emerge as a better choice.

E.2 Synthetic Setting

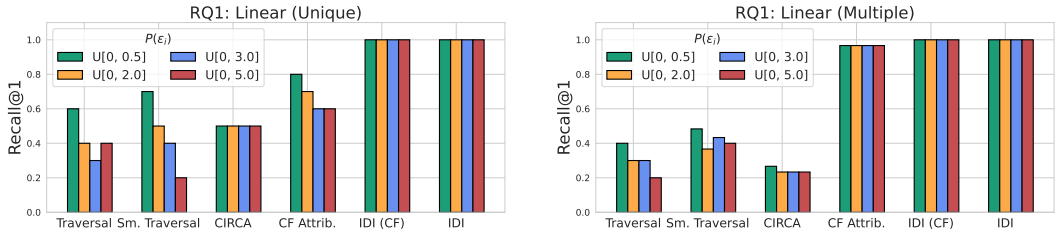
In this subsection, we evaluate the impact of ϵ_i variance on other datasets used in our study.

Petshop. For the real-world Petshop dataset, the exogenous variables ϵ are latent, preventing us from characterizing or controlling their variance. So we cannot experiment with this dataset.

All the synthetic experiments outlined below use 100 i.i.d. training samples to learn the SCM $\widehat{\mathcal{S}}$.

Linear SCM. In the main paper, we conducted experiments with each ϵ_i sampled from $U[0, 1]$. Here, we explore broader distributions by sampling from $U[0, b]$, with $b \in \{0.5, 2, 3, 5\}$. To reduce clutter, we focus on the best-performing baselines from the main results. Figure 7 presents results for the Linear SCM. The left panel corresponds to unique root cause test cases, while the right panel shows multiple root cause test cases, with Recall@1 on the Y-axis. As expected, in the linear setting,

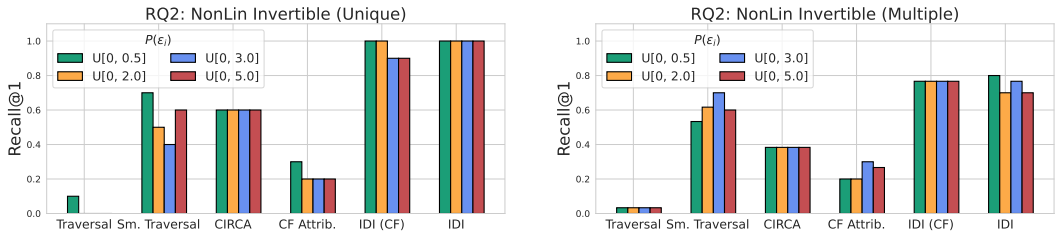
both interventional and counterfactual variants of IDI achieve Recall of 1 across all variance settings, with the CF Attribution baseline standing out as a strong competitor.



(a) Unique Root Cause (b) Multiple Root Causes

Figure 7: Linear Oracle SCM

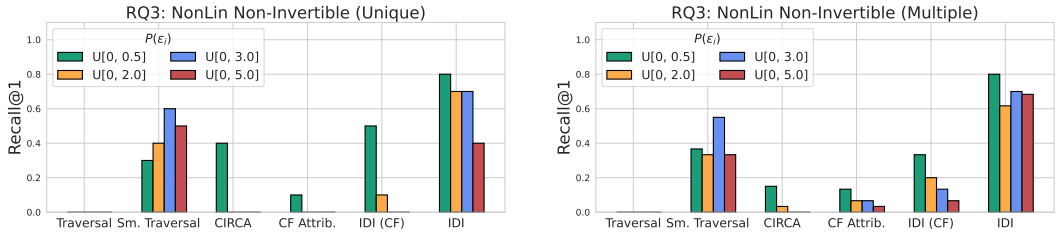
Non-Linear Invertible. Figure 8 presents results for this setting. For unique root cause test cases, IDI slightly outperforms the IDI (CF) variant of our approach. Among the baselines, CIRCA and Smooth Traversal emerge as strong competitors, while CF Attribution performs poorly at high variance. For multiple root causes, IDI falls slightly short of IDI (CF) in high variance scenarios, likely due to the overfitting or unstable training of the SCM $\widehat{\mathcal{S}}$. We infact observed high validation errors during training for high variances ϵ_is.



(a) Unique Root Cause (b) Multiple Root Causes

Figure 8: Non Linear Invertible Oracle SCM

Non-Linear Non-Invertible. This setting is the most challenging among all datasets because ϵ_i is not identifiable, causing CF approaches to struggle. As shown in Fig. 9, both CF Attribution and our IDI (CF) variant perform poorly, often achieving near-zero recall at high variance. While IDI also experiences performance drops compared to previous datasets, it remains comparatively robust when compared against the baselines. IDI stands out as the best approach across all variance settings in this dataset.



(a) Unique Root Cause (b) Multiple Root Causes

Figure 9: Non Linear Non Invertible Oracle SCM

F SENSITIVITY ANALYSIS OF THE ANOMALY THRESHOLD

We conducted this experiment on the Petshop dataset. In our main paper, we used a default anomaly threshold of 5. In this section, we assess the impact of baselines that implement the anomaly condition and IDI under varying anomaly thresholds, τ_i. For Z-Score, the threshold determines how many standard deviations a sample must deviate from the mean to be considered anomalous. We experimented with thresholds ranging from 2 to 7 and report the results in Table 9. Overall, we observed that IDI and the other baselines remained robust across different threshold choices, with IDI only showing performance degradation at a threshold of 2. We acknowledge that tuning the threshold is important in practice. However to tune it, we abnormal root cause test samples during training, since most samples in D_{trn} are non-anomalous. In the absence of such abnormal test samples, specifying this hyperparameter involves domain expertise.

	Recall@1						Recall@3					
	2	3	4	5	6	7	2	3	4	5	6	7
Traversal	0.73	0.80	0.87	0.90	0.87	0.80	0.73	0.83	0.87	0.90	0.87	0.80
Smooth Traversal	0.30	0.48	0.30	0.30	0.30	0.30	0.73	0.73	0.70	0.67	0.73	0.67
IDI (CF)	0.80	0.80	0.80	0.80	0.80	0.80	0.90	0.90	0.90	0.90	0.90	0.90
IDI	0.73	0.93	0.90	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93

Table 9: Results under variation of the anomaly detection threshold in the petshop dataset. We show both Recall@1 and Recall@3.

G SYNTHETIC CAUSAL GRAPHS

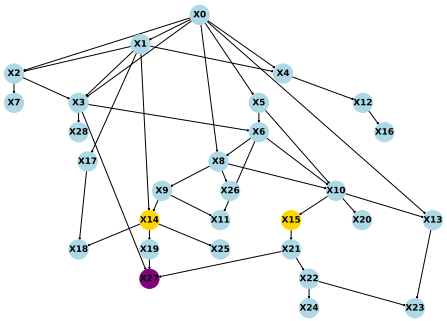


Figure 10: An example Synthetic test case

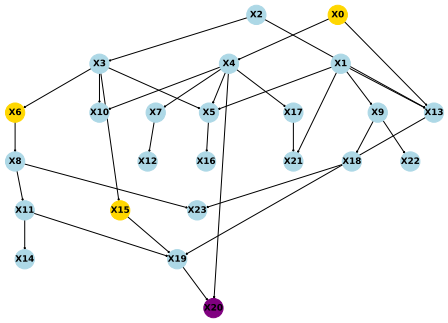


Figure 11: An example Synthetic test case

Figure 12: Random graphs sampled for our Synthetic Experiments

We show two instances of random graphs in Fig. 12. The purple node is the anomaly for which we need to find the root cause. The ground truth root cause nodes are shown in yellow. We typically observed that all nodes that are descendants of the yellow nodes also tend to exhibit anomalous behavior.