# PRIM: Meta-Learned Bayesian Root Cause Analysis

**Christopher Lohse**[1,2]*, **Anish Dhir**[3], **Amadou Ba**[2], **Bradley Eck**[2], **Marco Ruffini**[1], **Jonas Wahl**[4,5]

[1]University of Dublin, Trinity College, Dublin, Ireland
[2]IBM
[3]Gatsby Computational Neuroscience Unit, University College London
[4]Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Saarbrücken, Germany
[5]Department of Philosophy, University of Bergen, Norway

github.com/ChristopherLohse/PRIM

## Abstract

Root Cause Analysis (RCA) aims to identify the main drivers of anomalies in complex multivariate systems. The task is challenging due to the need for structural causal knowledge for some methods, and the computational cost of inference at test time. This paper introduces PRIM (**P**rior-fitted **R**oot cause **I**dentification with **M**eta-learning), a causal meta-learning approach that frames RCA as a Bayesian inference task over a prior of causal models. By marginalising out structural uncertainty, PRIM detects changes in the data-generating mechanism between normal and anomalous periods, infers distributional differences without explicit statistical testing, and implicitly learns causal structure without model fitting at test time. Following the simulation-based meta-learning paradigm of prior-fitted networks (PFNs), PRIM uses a transformer neural process (TNP) that jointly attends over normal and anomalous samples and the structural dependencies of nodes, enabling fast zero-shot inference in systems with up to 100 variables. Across synthetic benchmarks and two realistic benchmark datasets, PetShop and CausRCA, PRIM is competitive with other graph-free RCA methods.

## 1 Introduction

In complex multivariate dynamic systems, anomalies rarely occur alone. When a fault occurs, its effects propagate through the system, causing multiple variables to behave abnormally simultaneously. Root cause analysis (RCA) is the task of identifying the original source of an error based on observations of the system both before and during the anomalous period (Budhathoki et al., 2021). RCA is a critical task across many domains, including cloud microservices (Hardt et al., 2024; Shan et al., 2019; Pham et al., 2024), manufacturing (Mehling et al., 2026), medicine (Selvaraj and Piramanayagam, 2019; Lynn and Curry, 2011) as well as climate and sustainability (Jayswal et al., 2011; Belausteguigoitia, 2004). Existing approaches solve the RCA task either by assuming knowledge of a causal graph describing the dynamics of the system (Li et al., 2022; Budhathoki et al., 2022; Page et al., 1999) or by learning (parts of) a graph or statistical dependencies at inference time for example through causal discovery (Ikram et al., 2022). Graph-based methods require accurate prior knowledge sometimes not available in practice, while methods that infer the graph at test time face computational costs scaling superexponentially with the number of variables (Pham et al., 2025). Graph-free approaches based on statistical testing such as $\varepsilon$-diagnosis (Shan et al., 2019) and RCD (Ikram et al., 2022) come with the cost of a slower inference time, which scales superlinearly with the number of variables.

*Correspondence to Christopher Lohse: lohsec@tcd.ie.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

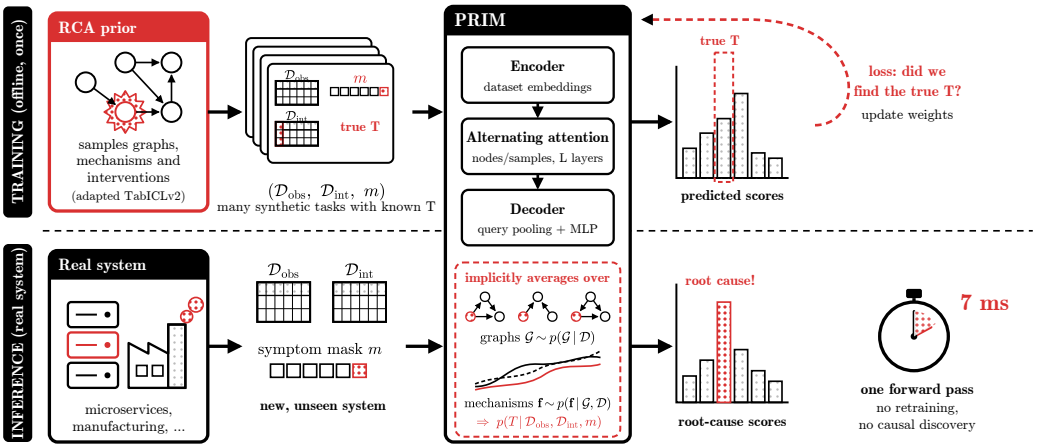



Figure 1: Simplified overview of the PRIM PFN training and inference process. The RCA prior samples synthetic data $(\mathcal{D}_{\text{obs}}, \mathcal{D}_{\text{int}}, m)$ with a known root cause $T$, and PRIM is trained until convergence. For a new real system, PRIM takes the observed data and symptom mask $m$ and returns root-cause scores in a single forward pass, implicitly averaging over the unknown graph $\mathcal{G}$ and mechanisms $\mathbf{f}$.

A Bayesian, TNP (Transformer Neural Process)-based approach addresses these issues directly. Bayesian model-averaging over graph and functional mechanism uncertainty mitigates the finite-sample and identifiability issues that affect point-estimate causal discovery (Dhir et al., 2025a; Lorch et al., 2021; Cundy et al., 2021; Robertson et al., 2025) and attends to functional uncertainty over multiple models. The TNP collapses the usual two-step causal-discovery-then-RCA pipeline into a single inference step, removing the need for explicit distribution-shift testing (Shan et al., 2019; Ikram et al., 2022; Hardt et al., 2024) and thus speeding up inference. This motivates our re-framing of RCA as a Bayesian causal meta-learning problem (Dhir et al., 2025a,b; Robertson et al., 2025), following the simulation-based paradigm of prior-fitted networks (PFNs) (Hollmann et al., 2023; Müller et al., 2022; Qu et al., 2026) which has proven effective for classification, anomaly detection (Hollmann et al., 2025), and time series forecasting (Hoo et al., 2025) with good results on real data (Hollmann et al., 2025; Qu et al., 2026). Figure 1 shows a simplified overview of the Bayesian TNP for RCA.

**Contributions** Our contributions are fourfold:

1. Inspired by Dhir et al. (2025b); Hollmann et al. (2023); Robertson et al. (2025), we design and implement PRIM (**P**rior-fitted **R**oot cause **I**dentification with **M**eta-learning), a zero-shot inference method that ranks root causes given normal and anomalous data and does not require structural knowledge of the data-generating system as an input.
2. To train PRIM, we adapt the TabICLv2 prior (Qu et al., 2026) to the root cause analysis task. To generate anomalies, we complement TabICLv2 with a variety of structural interventions that are selected according to a two-sample test to ensure appropriate intervention strength.
3. We discuss theoretical identifiability of root causes, adapting results by Squires et al. (2020); Jaber et al. (2020) to our setting.
4. We empirically validate that PRIM converges to the true root cause given sufficient interventional/anomaly data, and evaluate PRIM on several synthetic setups and two realistic benchmarks, PetShop (Hardt et al., 2024) and CausRCA (Mehling et al., 2026). PRIM exhibits robust general performance across well-supported settings.

## 2 Preliminaries

**Structural Causal Models** A Structural Causal Model (SCM) quantitatively represents causal relationships among multiple random variables and describes how changes in a system variable $X$

lead to changes in other variables (Pearl, 2009). In an SCM, each variable $X$ is assigned a value $X := \phi_X(\mathrm{pa}_X, \epsilon_X)$ through a function $f$ of its *causal parents* $\mathrm{pa}_X$ and an independent noise term $\epsilon_X$ (Pearl, 1995). Qualitative causal relationships in an SCM can be represented graphically as a Directed Acyclic Graph (DAG) $\mathcal{G}$ with directed edges $Z \to X,\ Z \in \mathrm{pa}_X$. *Causal discovery* is the task to infer the structure of $\mathcal{G}$ from observational or interventional data (Spirtes and Zhang, 2016).

**Root Cause Analysis** Consider variables $X = (X_1, \ldots, X_k)$ whose causal interactions are given by an SCM with underlying DAG $\mathcal{G} = (V, E)$. Following Budhathoki et al. (2021); Ikram et al. (2022) we define a **root cause** as a soft intervention on a subset of variables indexed by $T \subseteq \{1, \ldots, n\}$, resulting in a shifted distribution $P_X^T = \prod_{j \in T} \tilde{P}_{X_j|\mathrm{pa}_j} \prod_{j \notin T} P_{X_j|\mathrm{pa}_j}$, where $\tilde{P}_{X_j|\mathrm{pa}_j}$ represents the modified conditional distribution for intervened variables. Following Eberhardt and Scheines (2007), we distinguish between intervention types: a *hard intervention* $do(X_i = x)$ replaces the mechanism $f_i$ with a constant, eliminating the influence of $\mathrm{pa}(X_i)$. In contrast, a *soft intervention* replaces $f_i$ with a modified mechanism $f_i'$, altering the conditional distribution $P(X_i \mid \mathrm{pa}(X_i))$ without removing the causal edges from its parents. The *root cause analysis task* is to recover the intervention target set $T$ given observational samples $\mathcal{D}^{\text{obs}} = \{x^{(k)}\}_{k=1}^{n_{\text{obs}}} \sim P_X$ (normal operating conditions), interventional samples $\mathcal{D}^{\text{int}} = \{x^{(k)}\}_{k=1}^{n_{\text{int}}} \sim P_X^T$ (anomalous conditions under intervention), and one or more observed symptom(s) $m \in \{0,1\}^K$ with $m_j = 1$ if the node $j$ is anomalous. In our setup of RCA we consider that an observed anomaly/anomalies $m$ is known, which restricts the set of valid root causes $T$. Typically $m$ only contains one anomalous node, the 'problem' or 'target' node (Hardt et al., 2024), but it some case it can also be several 'alarm nodes'(Mehling et al., 2026).

Throughout this paper we assume that we have access to $m$, which is the case in several RCA benchmark settings (Hardt et al., 2024; Mehling et al., 2026). Each sample $x^{(k)} = (x_1^{(k)}, \ldots, x_n^{(k)})$ is a realization of the variables $X = (X_1, \ldots, X_n)$. In applied root cause analysis settings, the observational data corresponds to normal system behaviour, while the interventional data represents anomalous observations collected when the system is malfunctioning or degraded (Ikram et al., 2022; Assaad et al., 2023).

## 3 Bayesian Root Cause Analysis

Dhir et al. (2025a,b); Robertson et al. (2025) formalise causal discovery and inference as Bayesian meta-learning problems, training a neural process to approximate posteriors over causal structures from data. We adapt their formulation to RCA: rather than firstly inferring $\mathcal{G}$, our goal is to infer the root cause set $T$ directly, avoiding a two-step pipeline of causal discovery followed by RCA. An intervention on $T$ may induce a shift in functional relationships, noise distributions, or both, formalized as a change from the pre-intervention model $\mathbf{f}^{\text{obs}} = (\phi^{\text{obs}}, \epsilon^{\text{obs}})$ to the post-intervention model $\mathbf{f}^{\text{int}} = (\phi^{\text{int}}, \epsilon^{\text{int}})$. Here $\phi$ represents the collection of functional mappings and $\epsilon$ the collection of noise terms of the SCM. The change is localized to the intervention target: $\mathbf{f}_i^{\text{int}} = \mathbf{f}_i^{\text{obs}}$ for all $i \notin T$, while $\mathbf{f}_i^{\text{int}} \neq \mathbf{f}_i^{\text{obs}}$ for $i \in T$. A Bayesian formulation of RCA avoids committing to a single estimated DAG, propagating structural uncertainty about the DAG into the root cause posterior instead.

**Bayesian Causal Model** Adapting Dhir et al. (2025a,b) to the RCA setting, we define a *Bayesian causal model* (BCM) by drawing a DAG $\mathcal{G}$ from $\mathcal{G} \sim p_{\text{BCM}}(\mathcal{G})$. Conditional on $\mathcal{G}$, we sample the observational mechanism $\mathbf{f}^{\text{obs}} \sim p_{\text{BCM}}(\mathbf{f}^{\text{obs}} \mid \mathcal{G})$ and the root cause set $T \sim p_{\text{BCM}}(T \mid \mathcal{G})$. The interventional mechanism is then drawn as $\mathbf{f}^{\text{int}} \sim p_{\text{BCM}}(\mathbf{f}^{\text{int}} \mid T, \mathbf{f}^{\text{obs}}, \mathcal{G}) = \prod_{i \in T} p_{\text{BCM}}(\mathbf{f}_i^{\text{int}} | \mathcal{G}) \prod_{j \notin T} \mathbb{1}_{\mathbf{f}_j^{\text{int}} = \mathbf{f}_j^{\text{obs}}}$, reflecting that interventions modify only mechanisms or noises in $T$ and leave others invariant. The random variables $X^{\text{obs}}, X^{\text{int}}$ are distributed as $X^{\text{obs}} \sim p_{\text{BCM}}(X^{\text{obs}} \mid \mathbf{f}^{\text{obs}})$, respectively $X^{\text{int}} \sim p_{\text{BCM}}(X^{\text{int}} \mid \mathbf{f}^{\text{int}})$, and the corresponding datasets are generated as i.i.d. samples

$$\mathcal{D}^{\text{obs}} = \{X^{\text{obs}(k)}\}_{k=1}^{n_{obs}},\ X^{\text{obs}(k)} \sim X^{\text{obs}}, \qquad \mathcal{D}^{\text{int}} = \{X^{\text{int}(k)}\}_{k=1}^{n_{int}},\ X^{\text{int}(k)} \sim X^{\text{int}}.$$

We assume that $\mathrm{desc}(T) \subseteq \mathrm{anom}(X^{\text{obs}}, X^{\text{int}}) := \{\ell \in V \mid p_{\text{BCM}}(X_\ell^{\text{int}}) \neq p_{\text{BCM}}(X_\ell^{\text{obs}})\}$, meaning that each descendant of a root cause experiences a marginal distribution change[2]. This can be

[2]Observe that the converse inclusion $\mathrm{desc}(T) \supseteq \mathrm{anom}(X^{\text{obs}}, X^{\text{int}})$ holds by construction of our BCM.

considered a faithfulness assumption for root cause analysis as it precludes that the effect of the intervention cancels out along different paths to a descendant, 'coincidentally' reproducing the observational distribution. This assumption is implied by the $\mathcal{I}$-faithfulness assumption of (Squires et al., 2020), which is required for identifiability, see also Appendix C.2. As a consequence, this assumption allows us to sample the anomaly indicator $m$ uniformly from $\mathrm{desc}(T)$, $m \sim p(m \mid \mathrm{desc}(T))$.

**Bayesian Posterior and Identifiability of Root Causes** Defining $\mathbf{f} \equiv (\mathbf{f}^{\text{obs}}, \mathbf{f}^{\text{int}})$, the Bayesian posterior over the root causes depends solely on the mechanism change, making separate component specifications redundant. The factorization of the distribution of the Bayesian Model gives the following conditional independence statements: 1) $T \perp X^{\text{obs}}, X^{\text{int}} \mid \mathbf{f}, m, \mathcal{G}$: given the mechanism change, graph, and anomaly location, the data carry no additional information about $T$. (2) $\mathcal{G} \perp m \mid X^{\text{obs}}, X^{\text{int}}$, and (3) $\mathbf{f} \perp m \mid X^{\text{obs}}, X^{\text{int}}$: the anomaly indicator carries no additional information about the graph or mechanism once $X^{\text{obs}}, X^{\text{int}}$ are known. The Bayesian posterior over the root cause decomposes as

$$p(T \mid \mathcal{D}^{\text{obs}}, \mathcal{D}^{\text{int}}, m) = \sum_{\mathcal{G}} \int p(T \mid \mathbf{f}, \mathcal{G}, m)\, p(\mathbf{f} \mid \mathcal{G}, \mathcal{D}^{\text{obs}}, \mathcal{D}^{\text{int}})\, p(\mathcal{G} \mid \mathcal{D}^{\text{obs}}, \mathcal{D}^{\text{int}})\, d\mathbf{f}, \tag{1}$$

which marginalises over graph and functional mechanism uncertainty. See Appendix C.1) for the computation of this expression.

**Theorem 1** (Identifiability of Root Causes)**.** Assume that the data-generating SCM is causally sufficient with acyclic underlying DAG, and that mechanisms remain invariant for non-intervened nodes as in the definition of a BCM. Under $\mathcal{I}$-faithfulness (see Appendix C.2), $p_{\text{BCM}}(X_\ell^{\text{obs}})$ and $p_{\text{BCM}}(X_\ell^{\text{int}})$ together with the anomaly indicator $m$ identify the root-cause set $T$, even though the underlying DAG itself may not be uniquely identifiable.

This result is a direct adaptation of the results of Squires et al. (2020) (see also Jaber et al. (2020); Ikram et al. (2022) for parallel results) to our setting. A full proof for Theorem 1 is provided in Appendix C.2.

Bayesian model averaging is well-suited to this setting: it correctly propagates the residual graph ambiguity into the posterior over $T$. The posterior is intractable as the sum over $\mathcal{G}$ is exponential in $|V|$ and the integral over $\mathbf{f}$ is over an infinite-dimensional function space. Following Dhir et al. (2025b,a); Robertson et al. (2025), we therefore amortise inference with a prior fitted transformer neural process model $f_\theta$, trained on synthetic $BCM$ samples, that directly scores candidate root cause nodes

$$p_\theta(T = j \mid \mathcal{D}^{\text{obs}}, \mathcal{D}^{\text{int}}, m) = \mathrm{softmax}\big(f_\theta(\mathcal{D}^{\text{obs}}, \mathcal{D}^{\text{int}}, m)\big)_j, \quad j \in V. \tag{2}$$

Under standard assumptions on the mass concentration of the prior, posterior updates concentrate asymptotically on the true root cause set as the sample size goes to infinity (Ghosal and van der Vaart, 2017). The results of Hollmann et al. (2023); Qu et al. (2026); Dhir et al. (2025c) suggest empirically that TNPs are good approximators of Bayesian posterior distributions.

## 4 Related Work

**Transformer Neural Process and Prior-Fitted Networks** Since estimating the posterior over graph and functional uncertainty is hard, we use a Transformer Neural Process (TNP) trained as a Prior-Fitted Network (PFN), an approach proven effective on similar problems Hollmann et al. (2023, 2025); Dhir et al. (2025c,a); Robertson et al. (2025) across diverse domains Nguyen and Grover (2022); Feng et al. (2023); Ashman et al. (2024). PFN refers to the training paradigm: learning to approximate the posterior predictive by fitting to samples drawn from a synthetic prior. TNP refers to the architecture, a Neural Process built on the transformer (Vaswani et al., 2017). Trained this way, PFNs learn general tabular representations with strong performance on real world benchmarks, even though its trained on synthetic data on regression and classification Hollmann et al. (2023); Hoo et al. (2025); Müller et al. (2022), can be fine-tuned to new distributions Hoo et al. (2025), and as Neural Processes produce uncertainty-aware estimates at inference without intermediate steps such as explicit graph estimation (Dhir et al., 2025b). They have been applied to causal discovery Dhir

et al. (2025c) and effect estimation Dhir et al. (2025b); Robertson et al. (2025); in particular, Dhir et al. (2025b); Robertson et al. (2025) use a TNP to estimate interventional distributions given an intervention target, making it a natural foundation for estimating $T$ directly from $D^{\text{obs}}$, $D^{\text{int}}$, and $m$ without a separate causal discovery step. Using a Bayesian TNP avoids two-step algorithms that first fix a graph and then perform RCA, instead inferring $T$ jointly with $\mathcal{G}$ and the mechanisms. Further, it does not rely on statistical distribution difference tests to detect mechanism changes, which are often underpowered under limited interventional data.

**Root Cause Analysis** RCA closely aligns with intervention target estimation (Jaber et al., 2020; Varici et al., 2021, 2022; Yang et al., 2024; Squires et al., 2020): in both, interventions on a small set of nodes explain shifts in the overall distribution, and the goal is to identify these nodes. It has been applied to cloud microservices (Soldani and Brogi, 2022; Chen et al., 2014; Lin et al., 2018; Shan et al., 2019; Liu et al., 2021; Li et al., 2022; Pham et al., 2024; Liang et al., 2026; Hardt et al., 2024), manufacturing (Mehling et al., 2026), medicine (Selvaraj and Piramanayagam, 2019; Lynn and Curry, 2011), and sustainability (Jayswal et al., 2011; Belausteguigoitia, 2004). Existing approaches differ in whether they assume a known causal graph, and most impose some parametric assumption on the mechanism family.

*Graph-based* methods require the true causal graph as input. Traversal (Hardt et al., 2024) selects the highest-ranked anomalous node with no anomalous parent connected to $m$. CIRCA (Li et al., 2022) fits a linear Gaussian SCM and tests which conditional distribution has changed. Budhathoki et al. (2022) attribute root causes counterfactually via Shapley values; Nagalapatti et al. (2025) later replace the counterfactual estimates with interventional ones for improved robustness to out-of-distribution anomalies. PageRank (Page et al., 1999) and CausTR (Mehling et al., 2026) instead rank variables by graph proximity: PageRank by importance in the graph, and CausTR by causal proximity to alarm activations.

*Graph-free* methods relax this requirement, but typically introduce other assumptions. RCD (Ikram et al., 2022) performs local causal discovery under unknown soft interventions (Jaber et al., 2020; Squires et al., 2020), later extended to settings with partial structural knowledge (Ikram et al., 2025). RUN (Lin et al., 2024) and Han et al. (2025) instead recover a causal graph via neural Granger causal discovery from time series, combined with PageRank-based and integrated Granger-causal RCA, respectively. The remaining methods are graph-free by design: $\varepsilon$-diagnosis (Shan et al., 2019) uses two-sample tests, BARO (Pham et al., 2024) ranks by deviation magnitude, TimeRank (Mehling et al., 2026) ranks by recency of value changes, and Meta-RCA (Liang et al., 2026) combines LLMs with a Bayesian belief model and an LLM-derived knowledge graph.

PRIM differs from all the above method categories. In difference to graph-based methods such as Traversal, CIRCA, PageRank, CausTR and Shapley attribution, it requires no causal graph. In contrast to graph-free methods that estimate a graph at test time (RCD and RUN), its cost does not increase with the search over structures. Unlike methods that rely on explicit distribution-shift testing (CIRCA, $\epsilon$-diagnosis) or deviation heuristics (BARO, TimeRank, CausTR), PRIM's ranking is a single amortised posterior estimate. This retains a probabilistic treatment of graph and mechanism uncertainty via Bayesian model averaging, rather than committing to a point estimate at any stage.

## 5 Prior and Model Architecture

The following section outlines the design of our main contributions: adapting a prior of SCMs for the RCA setting and defining a transformer neural process architecture that learns to approximate the Bayesian posterior of root causes.

### 5.1 Adapting a prior for the RCA setting

Specifying a good prior is crucial for any PFN to work in real-life settings using the simulation-to-real approach (Hollmann et al., 2023; Qu et al., 2026). Unlike other priors, which are only used to do classification, regression, or causal inference (Hollmann et al., 2023; Qu et al., 2026; Dhir et al., 2025a); for root cause analysis the distribution shift in the anomalous data needs to resemble real-life failures in complex systems. We propose that a good prior for RCA must satisfy the following three criteria:

1. The underlying BCM must sample from a graph whose density and structure resemble the system to be estimated.
2. The mechanisms for the structural equations must resemble the underlying mechanisms observed during inference.
3. The intervention specification, i.e. the shift causing the shift in the root cause variable, must resemble the shifts we expect to see during inference.

The steps of the data sampling process for generating data from the prior is shown in Figure 2. To generate the causal Graph that underlies the data-generation process, we sample sparse DAGs (Erdős–Rényi, Barabási–Albert, or bipartite) over $K \sim \mathrm{Unif}(K_{\min}, K_{\max})$ nodes with expected in-degree $\bar{d} \sim \mathrm{Unif}(1.8, 2.5)$. We do so to satisfy the first point for a good RCA prior: as the systems usually observed for RCA are sparse, the in-degree is based on common values for real systems.

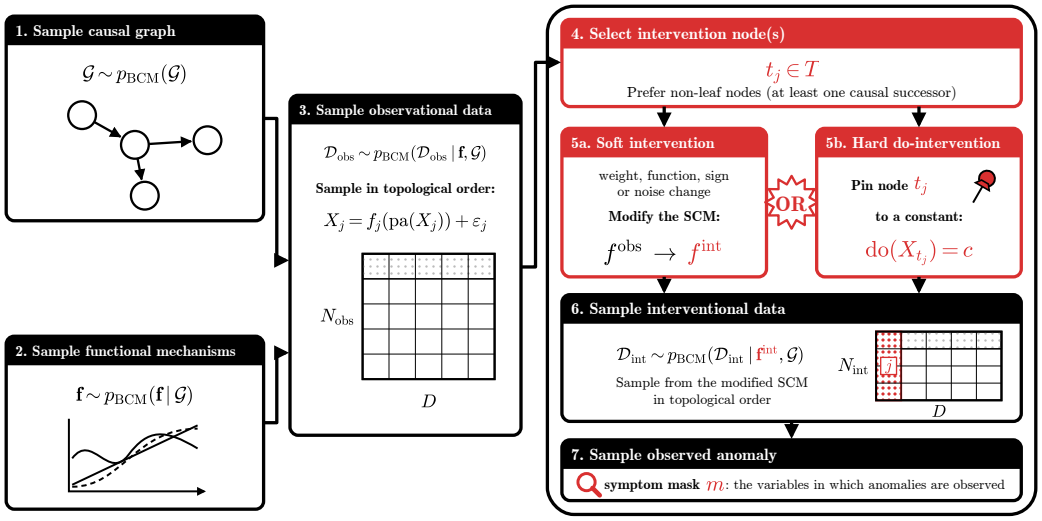



Figure 2: Overview of the data generation process for the prior. A causal graph $\mathcal{G}$ and functional mechanisms $\mathbf{f}$ are sampled to define an SCM. Observational data $\mathcal{D}_{\mathrm{obs}}$ is drawn by ancestral sampling, after which a target node $t_j \in T$ is selected and the SCM is perturbed via a soft intervention ($\mathbf{f}^{\mathrm{obs}} \rightarrow \mathbf{f}^{\mathrm{int}}$) or a hard do-intervention pinning $t_j$ to a constant. Interventional data $\mathcal{D}_{\mathrm{int}}$ is then sampled from the modified SCM. The final sample is $(D^{\mathrm{obs}}, D^{\mathrm{int}}, m)$.

As a starting point for generating the observational distribution to satisfy point two, we use the TabICLv2 Prior (Qu et al., 2026), since this prior design has shown promising performance on real data in earlier work (Qu et al., 2026), and offers a broad set of different distributions. We keep the functional mechanisms (mixed MLP, tree, linear, and Gaussian process) of TabICLv2, and also use its discrete data sampling. We further add temporal structure to the prior, with lower noise and higher autocorrelation than the other used distributions, as many RCA datasets are time series.

As our method takes as input the pre-anomaly data $\mathcal{D}^{\mathrm{obs}}$ as well as the anomalous data $\mathcal{D}^{\mathrm{int}}$, we need to specify the types and effects of interventions included in the prior. This is not included in the TabICLv2 prior, since it never samples interventional data. This is one of the contributions made by us. Following standard synthetic RCA data generation (Assaad et al., 2023), we differentiate between interventions that shift the weight of the structural equations, shift the mean, inflate the noise. We further add a fixed value do-intervention and sample 1 to $t_{max}$ (maximum number of parallel faults) root causes at a time. For higher realism, we also introduce sampling failures that are restricted to a local neighbourhood and do not affect the majority of the system.

To select an appropriate intervention strength, we run a Maximum Mean Discrepancy (MMD) two-sample permutation test (Gretton et al., 2012) across a grid of different intervention strengths. We select the smallest magnitude where the mean test rejection rate exceeds 0.9 at a significance level of $\alpha = 0.05$, ensuring that the induced distributional shift is detectable and learnable by any estimator. We randomly sample these types of interventions in each training pair generation, following our setup of the BCM defined in section 3. In addition, we sample multiple causally independent root

causes. We assume that the root cause $T$ is never a leaf node nor the observed anomaly $m$ itself. Each training samples samples $(\mathcal{D}^{obs}, \mathcal{D}^{int}, m)$ from the prior $p(\mathcal{M})$. Our prior satisfy our identifiability assumptions; for an in-depth description of the prior design, the data sampling process, and all hyperparameters, see Appendix D.1.

### 5.2 Model Architecture

Our model, PRIM (**P**rior-fitted **R**oot cause **I**dentification with **M**eta-learning), predicts which node of a system was intervened on (the root cause) from paired observational and interventional data, using an architecture drawing inspiration from the causal-structure-induction model of Ke et al. (2023). An overview of the architecture is given in Figure 3. Observational and interventional samples are each encoded into per-node embeddings by concatenating a data-type-specific value projection, a fixed sinusoidal node-identity code, a learned sample-type tag (observational/interventional), and a learned anomaly tag derived from an optional binary symptom mask $m$. A trainable per-node query token $z^{(0)}$ is appended to the resulting sequence, and $L$ pre-norm transformer blocks refine it via attention that alternates between attending across the $K$ nodes and attending across the $N$ samples (observational, interventional, and the query). This enables observational and interventional samples attend to each other bidirectionally at every layer. After the final layer norm, an attention layer reads out the evolved query $\bar{z}$ over the evolved samples $\bar{\mathbf{H}}$ to produce a per-node summary $\mathbf{u}$. A two-layer MLP decodes $\mathbf{u}$ into per-node logits $\hat{T} \in \mathbb{R}^k$.

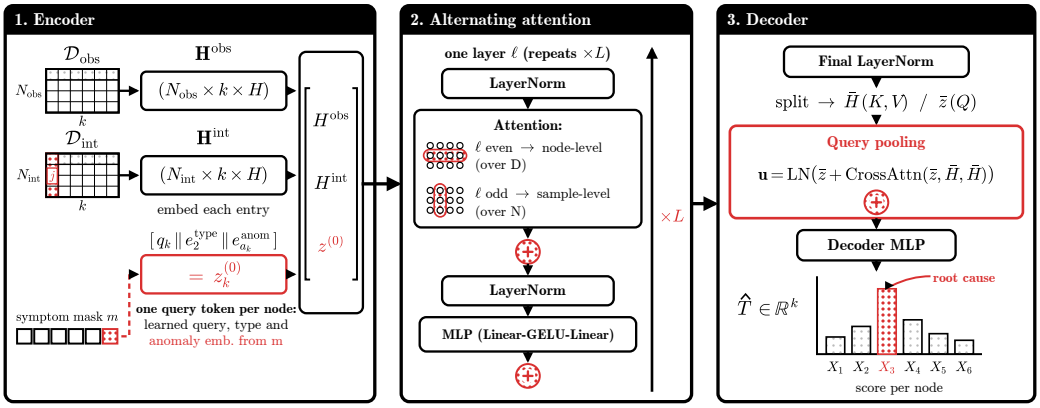



Figure 3: PRIM-NeuralProcess architecture (see text for description). $K$: number of nodes; $N$: number of samples in the sequence (observational + interventional + the query token); $z_k^{(0)} = [q_k \,\|\, e_2^{\text{type}} \,\|\, e_{a_k}^{\text{anom}}]$: learned per-node query token, with $a_k = m_k$ if the optional symptom mask $m$ is given, else a fixed "not provided" index; "+": residual (skip) connection; $\bar{z}, \bar{\mathbf{H}}$: query/samples after the final layer norm; $\mathbf{u}$: per-node summary produced by QueryPooling; $\hat{T} \in \mathbb{R}^k$: output logits.

The model is trained on synthetic SCMs drawn from $p(\mathcal{M})$ to predict the (possibly multiple) true root-cause nodes $T \subseteq \{1, \ldots, k\}$, minimizing the cross-entropy

$$\mathcal{L} = \mathbb{E}_{\mathcal{M}} \, \mathbb{E}_{T, \mathcal{D}|\mathcal{M}} \left[ -\sum_{i=1}^{k} y_i \log \hat{y}_i \right], \qquad y_i = \frac{\mathbf{1}[i \in T]}{|T|}, \tag{3}$$

where $\hat{y} = \text{softmax}(\hat{T})$ and $y$ is a uniform target distribution over the true root-cause node(s). This reduces to the standard one-hot cross-entropy when $|T| = 1$. An overview is shown in Figure 3; full architectural details are given in Appendix E.

## 6 Experiments

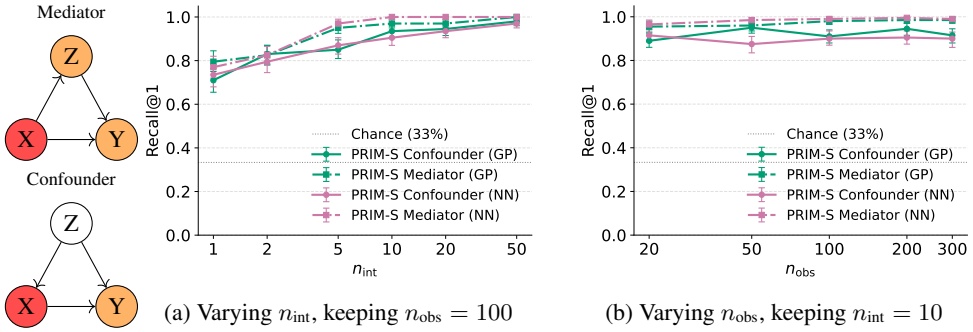
In order to evaluate PRIM we ran several experiments across simulated and realistic benchmarks. We first test for empirical convergence, then for different system sizes and finally for realistic data with unknown distributions. We use two different models for the experiment: PRIM-S a smaller model is

used for experiments with up to $k_{max} = 10$ and PRIM-L is used for the other experiments with up to $k_{max} = 100$ variables the detailed parameters are given in Appendix F.

### 6.1 Synthetic Data

In this section, we evaluate the performance of PRIM on RCA problems that resemble its training set, using GP and neural network mechanisms to cover a wide range of nonlinear functions. We report 90% bootstrap confidence intervals (based on 500 resamples) over 200 episodes.

**Empirical Convergence Study** In the first experiment, we evaluate in a controlled environment if PRIM correctly identifies the true root cause given sufficient interventional and observational data. This is an empirical sanity check of our analytical identifiability result. As we want to test a standard RCA setup we always assume fewer anomalous samples $n_{int}$ than normal samples $n_{obs}$. In Figure 4a, we test if, for a fixed $n_{obs} = 100$ and a varying $n_{int}$, more anomalous samples lead to convergence of the model to the true root causes, and the same for a fixed $n_{int} = 10$ (a realistic number of anomalous samples) by varying $n_{obs}$. We test on two three-node structures shown in Figure 4: a mediator and a confounder, with $X$ always representing the root cause. This tests whether PRIM correctly identifies the root cause in the presence of indirect causal paths and common causes. To generate the data, we use a Gaussian process and a neural network as structural functions. In the mediator setting (root cause is a root node), we intervene by inflating its noise, for both mechanisms. In the confounder setting (root cause has a parent), we instead scale the structural function generating it from its parent: its output scale for the Gaussian process, its weights for the Neural Network.



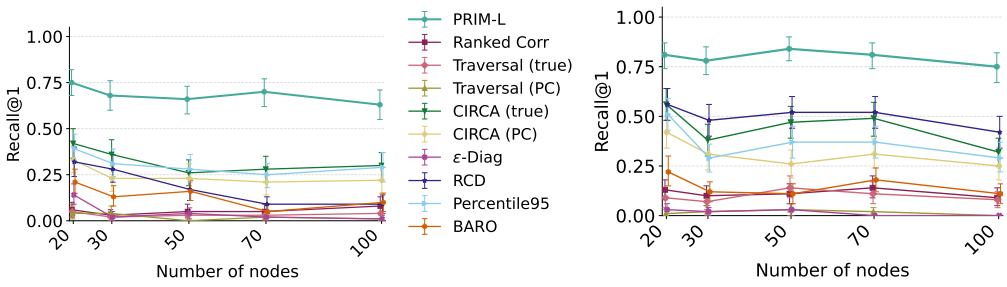

(a) Varying $n_{\text{int}}$, keeping $n_{\text{obs}} = 100$ (b) Varying $n_{\text{obs}}$, keeping $n_{\text{int}} = 10$

Figure 4: Three-node confounder vs. mediator scenario. X (■ red) is the root cause; downstream nodes (■ orange) are descendants; unaffected nodes are white.

Figure 4a shows that for both noise functions PRIM converges to the true root cause given enough interventional data. Figure 4b illustrates that more observational data does help the model to converge to the true root cause given a limited amount of $n_{int}$. We also tested this for a multi-RCA setting, and out-of-distribution generalisation with the same results: given enough interventional data, PRIM converges to the true root cause set. The results are provided in Appendix G.1 and G.2 respectively.

**Different System Sizes** To assess scalability, we evaluate PRIM across randomly sampled SCMs of varying size at $n_{\text{obs}} = 100$ and $n_{\text{int}} = 10$, with performance measured by Recall@1 ( Figure 5). We include noise distributions that are identical to the prior PRIM was trained on to assess that its ability to detect in distribution, these distribution might not fit some parametric/linearity assumption of some of the methods compared against. The intervention setup follows the calibrated prior described in Section 5.1: Weight-change and additive shift rescale or offset the node's output after its structural equation is evaluated, identically for the GP and the NN; hard do-intervention replaces that output with a constant; and noise inflation scales the exogenous noise term added inside the mechanism itself, on top of the NN's forward pass or the GP's sampled kernel value respectively.

We also include PC-versions of graph-based methods (PC-Traversal, PC-CIRCA), where the causal graph is estimated via the PC algorithm (Spirtes et al., 2000), providing graph-free variants of these methods for a fair comparison. Further, we add Percentile95 as a simple benchmark introduced by Fang et al. (2026), which ranks the root causes based on the maximum difference between anomalous samples and the 95 percentile of the distribution of the normal data.




(a) Gaussian Process Functions

(b) Neural Network Functions

Figure 5: Recall@1 vs. number of nodes at $n_{\text{obs}} = 100$, $n_{\text{int}} = 10$.

As highlighted in Figure 5, PRIM outperforms all baselines in both GP and NN settings, including methods with access to the ground truth graph, and its Recall@1 degrades more slowly with increasing graph size than any other method. Recovering $T$ from the GP data is harder across the board due to weaker distributional shifts compared to NN weight-change interventions. CIRCA is the strongest competitor in the GP setting, while for NN data CIRCA leads for smaller graphs (up to $\sim$20 nodes) and RCD becomes competitive at larger sizes. The 95Percentile simple benchmark performs surprisingly well on both settings, similar to CIRCA, which has access to the true graph, in the GP setting. The PC-based variants are substantially worse than their True-graph counterparts, likely due to nonlinear functional mechanisms that violate the assumptions of the executed version of the PC algorithm, even though they still outperform RCD in the GP and $\varepsilon$-diagnosis (which also assume linear relationships) in both settings. The full results over different interventional sizes and the small model can be found in Appendix G.3.

Since inference time is crucial for real-world deployment, we benchmarked all methods across node sizes. PRIM stays constant at 7ms (large model) as it requires only a single forward pass, while other methods scale exponentially. Full results are in G.4. On a MacBook Pro M3 Max, PRIM runs in $\sim$123ms on CPU and $\sim$17ms on MPS GPU across all node sizes.

### 6.2 Realistic Data with unknown Distribution

We evaluate on two real-world benchmarks spanning cloud microservices and manufacturing. The PetShop benchmark (Hardt et al., 2024) (CC Public License) is a microservice dataset based on the AWS Observability Workshop comprising $\sim$44 services monitored via latency and availability metrics, with 48 fault injection issues across four traffic scenarios, each with a single root cause. $\mathcal{D}^{\text{obs}}$ covers the normal baseline period ($|\mathcal{D}^{\text{obs}}| \in \{589, 1652\}$ depending on scenario) and $\mathcal{D}^{\text{int}}$ the anomaly period ($|\mathcal{D}^{\text{int}}| = 5$ five-minute snapshots); the symptom mask $m$ indicates the entry-point service (PETSITE). The CausRCA benchmark (Mehling et al., 2026) (Apache License 2.0) is a partially real, partially simulated factory system with 100 runs across 19 fault injection experiments in three subsystems (probe: 34, coolant: 25, hydraulics: 41), featuring multiple simultaneous root causes; all firing alarm nodes are included in $m$. On average $|\mathcal{D}^{\text{obs}}| = 196$ and $|\mathcal{D}^{\text{int}}| = 123$ (500 ms discretisation). Model weights are never updated on either benchmark; we use PRIM-L throughout. We only report the average values on these data sets here.

Table 1: Average accuracy of graph-not-given RCA methods on PetShop (top-3 recall, averaged over six traffic/metric scenarios) and CausRCA (MAP@3, averaged over three systems; *Full* and *Sub*graph settings shown separately). Results marked * are from the original papers Hardt et al. (2024); Mehling et al. (2026); – indicates not reported. **Bold** is the best result per row. † denotes our method.

| | | $\varepsilon$-diag | RCD | TimeRank | Baro | Corr | CausTR+PC | PRIM† |
|---|---|---|---|---|---|---|---|---|
| PetShop | | 0.10* | 0.39* | 0.15 | 0.08 | 0.78* | – | **0.79** |
| CausRCA | Full | 0.02 | 0.04 | 0.24 | 0.09 | 0.05 | – | **0.27** |
| | Sub | 0.11 | 0.24 | 0.60 | **0.75** | 0.33 | 0.57* | 0.61 |

As shown in Table 1 among methods that do not require a known causal graph, PRIM achieves the highest average recall, outperforming all graph-not-given baselines. Based on the reported MAP@3 in Table 1. PRIM is the best graph-not-given method on the full system and is slightly worse than Baro on the subsystem, giving the overall most balanced performance. A full set of the detailed results, including a comparison with graph-based methods can be found in Appendices G.5 and G.6. On both datasets PRIM shows a balanced performance in line with other methods which highlights its general application.

## 7 Conclusion

In this paper we reformulate root cause analysis as a causal meta-learning problem and introduce PRIM, a model that identifies root causes via a single forward pass through a transformer neural process. PRIM approximates a Bayesian model-averaged posterior over graph and functional mechanism uncertainty using synthetic priors, effectively performing a parameter-free Bayesian test at inference time without any statistical testing or model fitting on real data. We present a way to design a prior for RCA and a TNP architecture that is able to estimate the true posterior of root causes based on the learned prior. PRIM is competitive with RCA baselines across diverse datasets and domains without requiring a causal graph or retraining at deployment time. Our synthetic experiments empirically confirm that $\mathcal{D}^{\text{int}}$ and $m$ can be used to identify $T$: PRIM converges to the true root cause(s) given sufficient interventional data. On the realistic benchmarks PetShop Hardt et al. (2024) and CausRCA Mehling et al. (2026), PRIM is competitive with other graph-free RCA methods. Our paper highlights the usefulness of meta-learned PFNs for root cause analysis.

**Limitations** Our approach has several limitations. First, identifiability of $T$ rests on the assumptions stated in Appendix C.2: an acyclic SCM, causal sufficiency, mechanism invariance for non-targets (these are intrinsic to our modeling approach), $\mathcal{I}$-faithfulness, and the requirement that the observed anomaly is induced by the mechanism changes. Violations such as latent confounding or faithfulness violations compromise theoretical identifiability. Second, a misspecified prior can degrade accuracy by failing to capture the true RCA distribution as observed in the full causRCA example. Finally, PRIM cannot exploit potentially available (partial) knowledge of the ground-truth graph.

## Acknowledgments and Disclosure of Funding

We would like to thank the anonymous reviewers for their helpful feedback and a constructive discussion which we believe helped shping this work into a better paper.
C.L. was partilaly funded by the EcoTwin project, which has received funding from the European Commission's Horizon Europe programme under grant agreement No 101157369.
JW received support from the German Federal Ministry of Research, Technology, and Space (BMFTR) as part of the project MAC-MERLin (Grant Agreement No. 01IW24007) and from the European Regional Development Fund (ERDF) and the German Federal State of Saarland within the scope of (To)CERTAIN (Project ID EFRE-AuF-0000942).

## References

Matthew Ashman, Cristiana Diaconu, Junhyuck Kim, Lakee Sivaraya, Stratis Markou, James Requeima, Wessel P. Bruinsma, and Richard E. Turner. Translation equivariant transformer neural processes. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, Vienna, Austria, 2024. JMLR.org.

Charles K Assaad, Imad Ez-Zejjari, and Lei Zan. Root cause identification for collective anomalies in time series given an acyclic summary causal graph with loops. In *International conference on artificial intelligence and statistics*, pages 8395–8404. PMLR, 2023.

Juan Carlos Belausteguigoitia. Causal chain analysis and root causes: the giwa approach. *AMBIO: A Journal of the Human Environment*, 33(1):7–12, 2004.

Kailash Budhathoki, Dominik Janzing, Patrick Bloebaum, and Hoiyi Ng. Why did the distribution change? In *International Conference on Artificial Intelligence and Statistics*, pages 1666–1674. PMLR, 2021.

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, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2357–2369. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/budhathoki22a.html.

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

Chris Cundy, Aditya Grover, and Stefano Ermon. Bcd nets: Scalable variational approaches for bayesian causal discovery. *Advances in Neural Information Processing Systems*, 34:7095–7110, 2021.

Anish Dhir, Matthew Ashman, James Requeima, and Mark van der Wilk. A meta-learning approach to bayesian causal discovery. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 14158–14178, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/24faedc5853648d5857f2cf086a2acac-Paper-Conference.pdf.

Anish Dhir, Cristiana Diaconu, Valentinian Mihai Lungu, James Requeima, Richard E. Turner, and Mark van der Wilk. Estimating interventional distributions with uncertain causal graphs through meta-learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=IQlcfc40Ja.

Anish Dhir, Ruby Sedgwick, Avinash Kori, Ben Glocker, and Mark van der Wilk. Continuous bayesian model selection for multivariate causal discovery. In *Forty-second International Conference on Machine Learning*, 2025c. URL https://openreview.net/forum?id=zydNWJzoVd.

Frederick Eberhardt and Richard Scheines. Interventions and causal inference. *Philosophy of science*, 74(5):981–995, 2007.

Aoyang Fang, Songhan Zhang, Yifan Yang, Haotong Wu, Junjielong Xu, Xuyang Wang, Rui Wang, Manyi Wang, Qisheng Lu, and Pinjia He. Rethinking the evaluation of microservice rca with a fault propagation-aware benchmark. *Proceedings of the ACM on Software Engineering*, 3(FSE): 1605–1627, 2026.

Leo Feng, Hossein Hajimirsadeghi, Yoshua Bengio, and Mohamed Osama Ahmed. Latent bottlenecked attentive neural processes. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=yIxtevizEA.

Subhashis Ghosal and Aad van der Vaart. Consistency: General theory. In *Fundamentals of Nonparametric Bayesian Inference*, Cambridge Series in Statistical and Probabilistic Mathematics, chapter 6, pages 123–164. Cambridge University Press, 2017.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL http://jmlr.org/papers/v13/gretton12a.html.

Xiao Han, Saima Absar, Lu Zhang, and Shuhan Yuan. Root cause analysis of anomalies in multivariate time series through granger causal discovery. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=k38Th3x4d9.

Michaela Hardt, William Roy Orchard, Patrick Blöbaum, Elke Kirschbaum, and Shiva Kasiviswanathan. The petshop dataset — finding causes of performance issues across microservices. In Francesco Locatello and Vanessa Didelez, editors, *Proceedings of the Third Conference on Causal Learning and Reasoning*, volume 236 of *Proceedings of Machine Learning Research*, pages 957–978. PMLR, 01–03 Apr 2024. URL https://proceedings.mlr.press/v236/hardt24a.html.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*, 2023.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, January 2025. ISSN 1476-4687. doi: 10.1038/s41586-024-08328-6. URL https://www.nature.com/articles/s41586-024-08328-6.

Shi Bin Hoo, Samuel Müller, David Salinas, and Frank Hutter. From tables to time: Extending tabpfn-v2 to time series forecasting. *arXiv preprint arXiv:2501.02945*, 2025.

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.

Azam Ikram, Kenneth Lee, Shubham Agarwal, Shiv Kumar Saini, Saurabh Bagchi, and Murat Kocaoglu. Root cause analysis of failures from partial causal structures. In *The 41st Conference on Uncertainty in Artificial Intelligence*, 2025. URL https://openreview.net/forum?id=5HeODZrG9E.

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*, volume 33, pages 9551–9561. Curran Associates, Inc., 2020. URL https://papers.nips.cc/paper/2020/hash/6cd9313ed34ef58bad3fdd504355e72c-Abstract.html.

Abhishek Jayswal, Xiang Li, Anand Zanwar, Helen H Lou, and Yinlun Huang. A sustainability root cause analysis methodology and its application. *Computers & chemical engineering*, 35(12):2786–2798, 2011.

Nan Rosemary Ke, Silvia Chiappa, Jane X Wang, Jorg Bornschein, Anirudh Goyal, Melanie Rey, Theophane Weber, Matthew Botvinick, Michael Curtis Mozer, and Danilo Jimenez Rezende. Learning to induce causal structure. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=hp_RwhKDJ5.

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*, KDD '22, page 3230–3240, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539041. URL https://doi.org/10.1145/3534678.3539041.

Shuai Liang, Pengfei Chen, Bozhe Tian, Gou Tan, Maohong Xu, Youjun Qu, Yahui Zhao, Yiduo Shang, and Chongkang Tan. MetaRCA: A Generalizable Root Cause Analysis Framework for Cloud-Native Systems Powered by Meta Causal Knowledge, March 2026. URL http://arxiv.org/abs/2603.02032. arXiv:2603.02032 [cs].

Cheng-Ming Lin, Ching Chang, Wei-Yao Wang, Kuang-Da Wang, and Wen-Chih Peng. Root cause analysis in microservice using neural granger causal discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 206–213, 2024.

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*, pages 3–20. Springer, 2018.

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

Lars Lorch, Jonas Rothfuss, Bernhard Schölkopf, and Andreas Krause. Dibs: Differentiable bayesian structure learning. *Advances in Neural Information Processing Systems*, 34:24111–24123, 2021.

Lawrence A Lynn and J Paul Curry. Patterns of unexpected in-hospital deaths: a root cause analysis. *Patient safety in surgery*, 5(1):3, 2011.

Carl Willy Mehling, Sven Pieper, Tobias Lüke, Julius Döbelt, and Steffen Ihlenfeldt. Enabling Joint Benchmarking of Automated Root Cause Analysis and Causal Discovery in Manufacturing Using the *causRCA* Dataset. *Procedia CIRP*, 139:114–120, January 2026. ISSN 2212-8271. doi: 10.1016/j.procir.2025.09.010. URL https://www.sciencedirect.com/science/article/pii/S221282712500976X.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=KSugKcbNf9.

Lokesh Nagalapatti, Ashutosh Srivastava, Sunita Sarawagi, and Amit Sharma. Robust root cause diagnosis using in-distribution interventions. In *International Conference on Learning Representations*, volume 2025, pages 25738–25765, 2025.

Tung Nguyen and Aditya Grover. Transformer neural processes: Uncertainty-aware meta learning via sequence modeling. *ArXiv*, abs/2207.04179, 2022. URL https://api.semanticscholar.org/CorpusID:250340974.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford infolab, 1999.

Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

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

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, July 2024. ISSN 2994-970X. doi: 10.1145/3660805.

Luan Pham, Hongyu Zhang, Huong Ha, Flora Salim, and Xiuzhen Zhang. Rcaeval: A benchmark for root cause analysis of microservice systems with telemetry data. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 777–780, 2025.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A better, faster, scalable, and open tabular foundation model. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=SxsyLjIfWB.

Jake Robertson, Arik Reuter, Siyuan Guo, Noah Hollmann, Frank Hutter, and Bernhard Schölkopf. Do-PFN: In-context learning for causal effect estimation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=0aNbl9b56B.

Suganya Selvaraj and Shanmughavel Piramanayagam. Impact of gene mutation in the development of parkinson's disease. *Genes & diseases*, 6(2):120–128, 2019.

Huasong Shan, Yuan Chen, Haifeng Liu, Yunpeng Zhang, Xiao Xiao, Xiaofeng He, Min Li, and Wei Ding. $\epsilon$-Diagnosis: Unsupervised and real-time diagnosis of small-window long-tail latency in large-scale microservice platforms. In *The World Wide Web Conference*, WWW '19, page 3215–3222, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450366748. doi: 10.1145/3308558.3313653. URL https://doi.org/10.1145/3308558.3313653.

Jacopo Soldani and Antonio Brogi. Anomaly detection and failure root cause analysis in (micro) service-based cloud applications: A survey. *ACM Comput. Surv.*, 55(3), February 2022. ISSN 0360-0300. doi: 10.1145/3501297. URL https://doi.org/10.1145/3501297.

Peter Spirtes and Kun Zhang. Causal discovery and inference: concepts and recent methodological advances. In *Applied informatics*, volume 3, pages 1–28. Springer, 2016.

Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000.

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

Burak Varici, Karthikeyan Shanmugam, Prasanna Sattigeri, and Ali Tajer. Scalable intervention target estimation in linear models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 1494–1505. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/0b94ce08688c6389ce7b68c52ce3f8c7-Paper.pdf.

Burak Varici, Karthikeyan Shanmugam, Prasanna Sattigeri, and Ali Tajer. Intervention target estimation in the presence of latent variables. In James Cussens and Kun Zhang, editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 2013–2023. PMLR, 01–05 Aug 2022. URL https://proceedings.mlr.press/v180/varici22a.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Karren Yang, Abigail Katcoff, and Caroline Uhler. Characterizing and learning equivalence classes of causal dags under interventions. In *International Conference on Machine Learning*, pages 5541–5550. PMLR, 2018.

Yuqin Yang, Saber Salehkaleybar, and Negar Kiyavash. Learning unknown intervention targets in structural causal models from heterogeneous data. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 3187–3195. PMLR, 02–04 May 2024. URL https://proceedings.mlr.press/v238/yang24d.html.

Yujia Zheng, Biwei Huang, Wei Chen, Joseph Ramsey, Mingming Gong, Ruichu Cai, Shohei Shimizu, Peter Spirtes, and Kun Zhang. Causal-learn: Causal discovery in python. *Journal of Machine Learning Research*, 25(60):1–8, 2024.

## A Changes in the Camera-Ready Version

Based on the discussions and reviews on openreview we made the following changes in the camera ready version and reran all experiments:

1. We improved the various unclear sections in the literature review and background sections.
2. We adapted the construction of the prior and clarified every step in the selection process of the prior, as requested by the reviewers.
3. We changed the model architecture slightly, allowing attention between the interventional and the observational data instead of only allowing information flow in one direction only. We further added a query token to ensure the mask is properly encoded and the model is positionally invariant. Instead of using a mean pooling and modelling an explicit difference between the distributions, we use an attention pooling. These changes led to performance improvements in our experiments. Additionally, these changes move the architecture further away from that of Dhir et al. (2025b), addressing a comment of the reviewers. We added the simple baseline of Fang et al. (2026) as a comparison algorithm, as suggested by reviewer *zCDN*.
4. We removed the finetuning experiments in the main experimental section, since this was a major point of criticism by all reviewers and caused some confusion. By adapting the prior and the architecture, our results match the ones of our initially finetuned models now.
5. We moved the multi-root cause identification experiment to the appendix.
6. We present the result tables for PetShop and causRCA more compactly and moved the full results to the appendix. We have run all but the CausTR+PC algorithms (specialised for CausRCA) on both data sets now and report the results. We only report the performance of graph-free algorithms in the main body now, as the joint reporting of graph-free and graph-aware algorithms caused confusion among the reviewers.

The initial submission of this work which contains the results of the old model architecture and prior, is available on arXiv.

## B Reproducibility

We publish the code with the Paper. For all the experiments and evaluation we used a fixed random seed to ensure reproducibility. The synthetic prior used to train PRIM, including graph sampling, structural equation families, exogenous noise, availability mode, and intervention types and calibration, is described in Section 5.1 and specified in full detail in Appendix D.1 and Appendix D.2, with calibrated intervention-strength ranges given in Table D.1. The padding and masking strategy used to handle variable graph sizes is described in Appendix D.3. Full architectural details of PRIM, including the data encoding, the alternating-attention transformer blocks, the query-pooling decoder, and the training objective, are given in Appendix E. All hyperparameters for training and the experiments, the model used per experiment, and compute requirements are listed in Appendix F (Tables F.1–F.3). Details of the PetShop and CausRCA benchmarks, including data splits and preprocessing, are given in Section 6.2 of the main text, with additional results in Appendices G.5 and G.6. For more explanations on these data sets we refer to Hardt et al. (2024) for the PetShop and Mehling et al. (2026) for the CausRCA data. Baselines marked with an asterisk in our tables are numbers reported directly by the original papers; all other baseline numbers are computed by our own code, calling the original authors' implementations rather than reimplementing them where available: Traversal, CIRCA, $\varepsilon$-diagnosis, RCD, random-walk, and counterfactual attribution use the implementations bundled with the PetShop benchmark Hardt et al. (2024); Baro uses RCAEval's Pham et al. (2025) own implementation; TimeRank (for PetShop, where it is not reported by the original authors) uses causRCA's Mehling et al. (2026) own time-recency baseline, adapted to PetShop's continuous telemetry via a threshold-based change-detection step we describe in code; the PC-discovered graph variants use `causal-learn`'s (Zheng et al., 2024) PC implementation; ranked-correlation and the percentile baseline are our own, graph-free implementations. Both the PetShop and CausRCA evaluations, including all baselines, can be run end-to-end on a laptop CPU with no GPU required (verified in under two minutes each), letting readers verify our reported numbers without access to a GPU cluster. We provide the code for that in the supplemental.

# C Derivations and Identifiability of Root Causes

## C.1 Derivation of the Bayesian Posterior

Since the components of $\mathbf{f} \equiv (\mathbf{f}^{\text{obs}}, \mathbf{f}^{\text{int}})$ are not independent, we for convenience do not model both functions separately in the posterior. The full distribution we want to estimate is:

$$p(T, \mathbf{f}, \mathcal{G} | \mathcal{D}^{\text{obs}}, \mathcal{D}^{\text{int}}, m) \tag{C.1}$$

by marginalising out $\mathcal{G}$ we get

$$p(T, \mathbf{f} | \mathcal{D}^{\text{obs}}, \mathcal{D}^{\text{int}}, m) = \sum_{\mathcal{G}} p(T, \mathbf{f}, \mathcal{G} | \mathcal{D}^{\text{obs}}, \mathcal{D}^{\text{int}}, m) \tag{C.2}$$

by additionally marginalising out $\mathbf{f}$ we get

$$p(T | \mathcal{D}^{\text{obs}}, \mathcal{D}^{\text{int}}, m) = \int \left( \sum_{\mathcal{G}} p(T, f, \mathcal{G} | \mathcal{D}^{\text{obs}}, \mathcal{D}^{\text{int}}, m) \right) d\mathbf{f} \tag{C.3}$$

which we can rewrite using the chain rule to

$$\sum_{\mathcal{G}} \int p(T \mid \mathbf{f}, \mathcal{G}, \mathcal{D}^{\text{obs}}, \mathcal{D}^{\text{int}}, m)\, p(\mathbf{f} \mid \mathcal{G}, \mathcal{D}^{\text{obs}}, \mathcal{D}^{\text{int}}, m)\, p(\mathcal{G} \mid \mathcal{D}^{\text{obs}}, \mathcal{D}^{\text{int}}, m)\, d\mathbf{f} \tag{C.4}$$

The factorization of the distribution of the Bayesian Model gives the the following conditional independence statements:

1. **Sufficiency of the mechanism, location, and graph for $T$**

   $$p\big(T \mid X^{\text{obs}}, X^{\text{int}}, \mathbf{f}, m, \mathcal{G}\big) = p\big(T \mid \mathbf{f}, m, \mathcal{G}\big) \tag{C.5}$$

   If we have the graph $\mathcal{G}$, the mechanism change $\mathbf{f}$ and the location of the observed anomaly the data does not carry any more information about the root cause $T$.

2. **Irrelevance of the anomaly indicator to the graph**

   $$p\big(\mathcal{G} \mid X^{\text{obs}}, X^{\text{int}}, m\big) = p\big(\mathcal{G} \mid X^{\text{obs}}, X^{\text{int}}\big) \tag{C.6}$$

   By definition of our Bayesian model the anomaly indicator $m$ does not carry any information about the graph; this means the graph is determined by the data alone the observed anomaly does not tell anything about $\mathcal{G}$ since it only occurs after the graph is already known/sampled.

3. **Irrelevance of the anomaly indicator to the mechanism**

   $$p\big(\mathbf{f} \mid \mathcal{G}, X^{\text{obs}}, X^{\text{int}}, m\big) = p\big(\mathbf{f} \mid \mathcal{G}, X^{\text{obs}}, X^{\text{obs}}\big) \tag{C.7}$$

   In a similar way the functional difference $\mathbf{f}$ does not depend on where the anomaly was observed. This can intuitively be explained as the observed anomaly only happens after the functional shift/ root cause of the anomlay has occurred.

We can therefore rewrite this as

$$p(T | \mathcal{D}^{\text{obs}}, \mathcal{D}^{\text{int}}, m) = \sum_{\mathcal{G}} \int p(T \mid \mathbf{f}, \mathcal{G}, m)\, p(\mathbf{f} \mid \mathcal{G}, \mathcal{D}^{\text{obs}}, \mathcal{D}^{\text{int}})\, p(\mathcal{G} \mid \mathcal{D}^{\text{obs}}, \mathcal{D}^{\text{int}})\, d\mathbf{f}. \tag{C.8}$$

### C.2 Identifiability of the Root Cause(s)

Here we provide our reasoning for why the posterior $p(T \mid \mathcal{D}^{\text{obs}}, \mathcal{D}^{\text{int}}, m)$ concentrates on the correct $T$ in the infinite-data limit. The argument is fully nonparametric and covers both soft and hard interventions. We focus on the multi-sample interventional setting, in which each interventional regime provides enough samples to identify its conditional distributions; this covers all our experiments except the single-anomaly CausRCA case. The argument relies on the identifiability result of Squires et al. (2020).

**Setting** We consider $K \geq 1$ interventional regimes. For each regime $k$, let $T^k_{\text{full}}$ denote the set of nodes whose mechanism is altered in regime $k$ relative to the observational regime, and write $T_{\text{full}} = \bigcup_{k=1}^{K} T^k_{\text{full}}$ for the union of all mechanism-changed nodes across regimes. The root cause set $T \subseteq T_{\text{full}}$ is the subset of mechanism-changed nodes that are causally responsible for the observed anomaly $m$, as defined in the main text.

**Definition C.1** ($\mathcal{I}$-Markov equivalence (Yang et al., 2018; Squires et al., 2020))**.** Two pairs of DAG and intervention target list, $(\mathcal{G}_1, T_{\text{full},1})$ and $(\mathcal{G}_2, T_{\text{full},2})$, are *$\mathcal{I}$-Markov equivalent* if they induce the same set of observational and interventional distributions over $X$, where each interventional regime preserves the conditional $p(X_i \mid \text{pa}_{\mathcal{G}}(i))$ for non-intervened nodes. The equivalence class of $(\mathcal{G}, T_{\text{full}})$ under this relation is the *$\mathcal{I}$-Markov equivalence class* ($\mathcal{I}$-MEC).

**Assumption C.1** (Acyclic SCM)**.** The data $\mathcal{D}^{\text{obs}}$ and $\mathcal{D}^{\text{int}} = \{\mathcal{D}^{\text{int},k}\}_{k=1}^{K}$ are generated by a structural causal model whose causal graph $\mathcal{G}$ is a directed acyclic graph over the observed variables $X = (X_1, \ldots, X_n)$.

**Assumption C.2** (Causal sufficiency)**.** There are no latent confounders.

**Assumption C.3** (Mechanism invariance for non-targets)**.** For each interventional regime $k$ and each $X_i \notin T^k_{\text{full}}$, the conditional $p(X_i \mid \text{pa}_{\mathcal{G}}(i))$ is invariant between the observational regime and regime $k$. Both soft and hard interventions on the targets in $T^k_{\text{full}}$ are permitted.

**Assumption C.4** ($\mathcal{I}$-faithfulness; Squires et al. (2020), Assumption 2)**.** The observational distribution is faithful to $\mathcal{G}$ in the usual sense, and conditional invariances across regimes correspond exactly to $d$-separations between intervention nodes and observed variables in the *augmented graph* $\mathcal{G}^{\mathcal{I}}$ (the graph obtained from $\mathcal{G}$ by adding, for each regime $k$, an intervention node $\zeta_k$ with edges $\zeta_k \to X_i$ for every $X_i \in T^k_{\text{full}}$). Intuitively, this rules out interventions that leave no statistical trace: every genuine mechanism change must produce a detectable difference in some conditional distribution, and effects from distinct interventions or distinct paths must not cancel exactly.

**Assumption C.5** (Anomaly variable)**.** The observed anomaly $m$ corresponds to a variable in $X$ whose anomalous behaviour is induced (directly or indirectly) by mechanism changes at a subset $T \subseteq T_{\text{full}}$ of intervened nodes. We refer to $T$ as the *root cause set*.

**Proof 1** (Proof of **Theorem 1**)**.** We first invoke the identifiability result of Squires et al. (2020). Their Assumption 2 (our Assumption C.4) states that for any regime $k$ and any disjoint $A, C$,

$$(A \perp\!\!\!\perp \zeta_k \mid C \cup \zeta_{[K]\setminus\{k\}})_{\mathcal{G}^{\mathcal{I}}} \quad \Longleftrightarrow \quad f^k(x_A \mid x_C) = f^{\text{obs}}(x_A \mid x_C).$$

The forward direction is the $\mathcal{I}$-Markov property (Assumptions C.1, C.2); the reverse is faithfulness (Assumption C.4), which rules out coincidental cancellations that would mask a genuine intervention.

Applying this with $A = \{i\}$ identifies the intervention targets, as in Squires et al. (2020, §3.2). A node $i$ lies in $T^k_{\text{full}}$ iff $\zeta_k$ and $i$ are adjacent in $\mathcal{G}^{\mathcal{I}}$, and adjacent nodes cannot be d-separated by any conditioning set. The biconditional above (Assumption C.4) translates this into the data-determined statement that no $C$ makes $f^k(x_i \mid x_C)$ and $f^{\text{obs}}(x_i \mid x_C)$ agree; for non-targets, conditioning on $\text{pa}_{\mathcal{G}}(i)$ does, by mechanism invariance (Assumption C.3). Hence $T^k_{\text{full}}$, and so $T_{\text{full}} := \bigcup_k T^k_{\text{full}}$, is identifiable, for both soft and hard interventions (Assumption C.3).

The set $T_{\text{full}}$ contains every node whose mechanism has changed in any regime, regardless of whether that change is causally relevant to the observed anomaly. Providing $m$ gives the following: a node $i \in T_{\text{full}}$ can only have contributed to the anomaly at $m$ if it lies upstream of $m$ in the causal graph (Assumption C.5). Mechanism changes at nodes that are not ancestors of $m$ propagate nothing to $m$ and are therefore irrelevant to the root cause analysis, even though they are genuine interventions. The root cause set is thus

$$T = \{i \in T_{\text{full}} : i \in \text{An}_{\mathcal{G}}(m)\}.$$

The same biconditional (Assumption C.4), applied with $A = \{m\}$, identifies $T$. Under causal sufficiency (Assumption C.2), $\zeta_k$ is d-connected to $m$ exactly when some target in $T^k_{\text{full}}$ is an ancestor of $m$ in $\mathcal{G}$ (Assumption C.1), so a node in $T_{\text{full}}$ is an ancestor of $m$ if no conditioning set makes the interventional and observational distributions of $m$ agree, a pattern determined by the data alone. Every DAG in the $\mathcal{I}$-MEC must therefore agree on which nodes in $T_{\text{full}}$ are ancestors of $m$, even if it disagrees about the rest of the ancestor relation. Marginalising over the $\mathcal{I}$-MEC yields the correct $T$ in the infinite-data limit, and residual ambiguities among nodes outside $T_{\text{full}}$ do not affect $T$.

**Compatibility of the prior with the assumptions** Our prior generation fulfils these assumptions by design: we sample only DAGs (no cycles) with no latent confounders, satisfying Assumptions C.1 and C.2. Soft interventions are modelled so as not to alter the graph structure, and hard interventions only drop the parents of the target node (more details in Appendix D.1); in both cases the conditionals of non-target nodes are unchanged, satisfying Assumption C.3. Because mechanism parameters are drawn from continuous distributions, the probability that an intervention is exactly cancelled by another intervention or by the functional form of a downstream mechanism is zero (a standard generic-position argument), so Assumption C.4 holds almost surely. Access to the anomaly variable is standard in RCA evaluations and is provided by our prior generation, satisfying Assumption C.5. Thus both the prior generation and the synthetic evaluation setups satisfy all the assumptions used above.

# D Data and Prior Design

## D.1 Data Generation

The more detailed data generation process is illustrated in Figure D.1. For each training instance, a DAG $\mathcal{G}$ and a set of structural equations $\mathbf{f}$ are sampled to form an SCM.

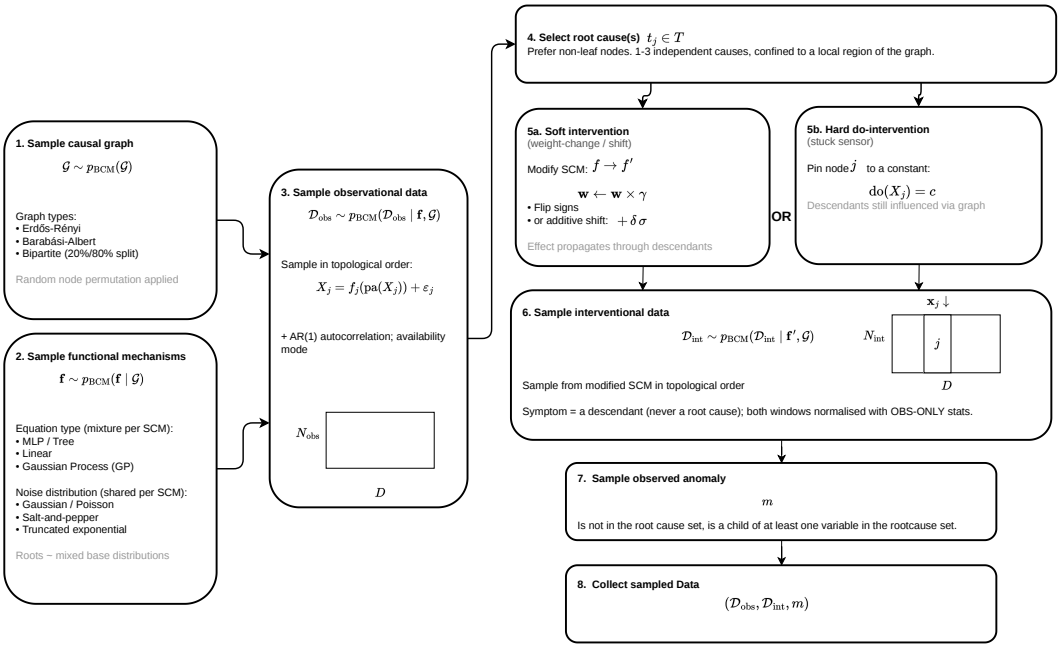



Figure D.1: Overview of the data generation process. A causal graph $\mathcal{G}$ and functional mechanisms $\mathbf{f}$ are sampled to define an SCM. Observational data $\mathcal{D}_{\mathrm{obs}}$ is drawn by ancestral sampling, after which a target node $t_j \in T$ is selected and the SCM is perturbed via a soft intervention ($\mathbf{f}^{\mathrm{obs}} \to \mathbf{f}^{\mathrm{int}}$) or a hard do-intervention pinning $t_j$ to a constant. Interventional data $\mathcal{D}_{\mathrm{int}}$ is then sampled from the modified SCM.

Observational data $\mathcal{D}_{\mathrm{obs}} \in \mathbb{R}^{N_{\mathrm{obs}} \times D}$ is collected by forward-sampling in topological order. Root-cause nodes $T$ are then selected and a soft or hard intervention is applied, yielding interventional data $\mathcal{D}_{\mathrm{int}} \in \mathbb{R}^{N_{\mathrm{int}} \times D}$. The triple $(\mathcal{D}_{\mathrm{obs}}, \mathcal{D}_{\mathrm{int}}, T)$ forms one training example. Note for all experiments except the finetuning for multiRCA and CausRCA $|T| = 1$ meaning we only inject one root cause.

## D.2 Prior over Causal Models

The prior design follows the design of Qu et al. (2026) with some specific changes to make it more suitable for the root cause analysis tasks such as different types of interventions, constant baselines and a rich range of noise distributions as well as only sampling sparse graphs.

**Graph structure** As the usual set-up for root cause analysis rarely includes dense causal graph we assume a sparse structure of the causal DAG. Each episode samples a DAG $\mathcal{G}$ over $K \sim \mathrm{Unif}(K_{\min}, K_{\max})$ nodes. The graph family is chosen uniformly from Erdős–Rényi (ER), Barabási–Albert (BA), and bipartite. Edge density is controlled via expected in-degree $\bar{d} \sim \mathrm{Unif}(1.8, 2.5)$, giving $p_{\mathrm{edge}} = \bar{d}/(K-1)$. A uniform node-index permutation removes topological-ordering bias.

**Structural equations** An episode draws one mechanism family per SCM from {MLP, tree, linear, GP}, with mixture weights $(0.50, 0.13, 0.22, 0.15)$. The MLP and tree mechanisms, the root-variable sampler, and the discretisation are taken from the TabICL prior (Qu et al., 2026). Follwing them we initialise small, random MLPs with block-sparse weights and injected per-layer noise, and tree ensembles (decision tree / random forest / extra-trees / gradient boosting) fit to random synthetic targets, yielding piecewise-constant, threshold-style mechanisms; see Qu et al. (2026) for details.

The linear mechanism is a simplified $X_k = \mathbf{w}_k^\top \tilde{X}_{\mathrm{pa}(k)} + \varepsilon_k$, and the Gaussian-process mechanism is unchanged (RBF/Matérn kernel, log-uniform lengthscale, Gamma-distributed observation noise), with an auxiliary $\mathcal{N}(0,1)$ input, independently resampled per node, appended to the kernel inputs so the mechanism's output is not purely additive in the observation noise.

Root nodes are sampled independently of the mechanism family, from a per-node distribution fixed once at initialisation (Normal / Uniform / Laplace / Student-$t$ / multimodal / mixed), following TabICL's `XSampler` (Qu et al., 2026).

**Availability mode** With probability $0.35$, an SCM instead mimics a near-constant failure metric: every node operates near a high baseline with tiny noise.

**Exogenous noise** Additionally we let each SCM draws its noise distribution uniformly from {Gaussian, Poisson, salt-and-pepper, truncated-exponential} with a low, log-uniform scale, and with probability $0.6$ applies AR(1) autocorrelation to mimic the slowly-varying signals typical of monitoring data. This enables our prior to work with time series data.

**Interventions/ Sampling of Root Causes** This is to mimic the failure of a part of the system enabling the pior to be used for root cause analysis. Each node in $T$ is intervened on independently with one of weight-change ($62\%$), additive shift ($25\%$), hard do-intervention ($8\%$), or noise inflation ($5\%$). Interventions act on the mechanism's *output*, so they apply identically across mechanism families and to root and non-root nodes ($\mu_k, \sigma_k$: observational mean/std of $X_k$; $s \sim \mathrm{Unif}\{-1,+1\}$):

- **Weight change:** $X'_k = \mu_k + s\gamma(X_k - \mu_k)$, $\gamma \sim \mathrm{Unif}(5,9)$, sign flipped w.p. $0.8$.
- **Additive shift:** $X'_k = X_k + s\delta\sigma_k$, $\delta \sim \mathrm{Unif}(1,4)$.
- **Hard do-intervention:** $X'_k = \mu_k + sm\sigma_k$, $m \sim \mathrm{Unif}(1,4)$, collapsing $\mathrm{Var}(X_k)$ to zero ("stuck sensor").
- **Noise inflation:** the exogenous noise scale is multiplied by $c \sim \mathrm{Unif}(3,6)$.

In availability mode, shift and do-intervention instead drop the metric by a fraction of the baseline. We then sample the observed anomli(es) $m$, which are children of the sampled root cause.

### D.3 Padding and Variable Graph Size

PRIM operates on graphs of varying size $k \le K_{\max}$, where $K_{\max}$ is the maximum number of nodes the model was trained on. To handle variable-size inputs with a fixed-size transformer, we follow the padding strategy of Hollmann et al. (2023, 2025).

**Data padding** For a graph with $k$ real nodes, both $\mathcal{D}^{\mathrm{obs}}$ and $\mathcal{D}^{\mathrm{int}}$ are zero-padded along the node dimension to size $K_{\max}$. Real node values are z-score normalised and then scaled by a factor of $K_{\max}/k$ before padding, so that the magnitude of activations remains consistent regardless of graph size. Padded positions are filled with zero. The normalisation is based on the observational data only.

**Attention masking** A binary padding mask is constructed such that positions $j \ge k$ are marked as padding. This mask is passed to all transformer layers, where it is used as a key padding mask in the node-level attention step, preventing padded positions from contributing to attention scores. After every LayerNorm operation, padded positions are explicitly re-zeroed by elementwise multiplication with the mask, preventing LayerNorm from reintroducing non-zero values at padding locations.

**Output masking** At inference time, logits at padding positions $j \ge k$ are set to $-\infty$ before the softmax, ensuring the model can only predict valid nodes as root causes. During training, the same masking is applied before the cross-entropy loss.

### D.4 Parameters of the intervention Strengths

For each intervention type we ran an RBF-kernel MMD two-sample permutation test (Gretton et al., 2012) ($\alpha = 0.05$, $n$=110 episodes per magnitude) between the observational and interventional windows, and set the lower bound of each range to the smallest tested magnitude at which the test's

rejection rate is $\gtrsim 0.9$. This means the weakest sampled fault is still reliably detectable. We show the values in Table D.1.

Table D.1: Calibrated intervention-strength ranges.

| Intervention | Mixture weight $\pi_k$ | Range | Reject rate @ floor | Notes |
|---|---|---|---|---|
| Weight-change | 0.62 | $\gamma \sim \mathrm{Unif}(5, 9)$ | 0.93 | flip w.p. 0.8 |
| Additive shift | 0.25 | $\delta \sim \mathrm{Unif}(1, 4)$ | 0.98 | |
| Hard do-intervention | 0.08 | $m \sim \mathrm{Unif}(1, 4)$ | 0.96 | |
| Noise inflation | 0.05 | $c \sim \mathrm{Unif}(3, 6)$ | – | not calibrated |

## E PRIM Architecture Details

PRIM (**P**rior-fitted **R**oot cause **I**dentification with **M**eta-learning), is built around an alternating node-/sample-level attention encoder together with an attention(Vaswani et al., 2017) based, Neural-Process-style readout.

Our architecture is oriented on the causal-structure-induction architecture of Ke et al. (2023).

However, Ke et al. (2023) infer a full graph structure from a set of observational and interventional samples. In contrast our problem setting is different: the model receives both observational (normal) and interventional (anomalous) data and predicts which node was intervened on: the root cause. An overview of the architecture is shown in Figure 3. Variable graph sizes are handled by zero-padding and attention masking following Hollmann et al. (2023, 2025). For this we provide implementation details in Appendix D.3.

**Data Encoding** Observational and interventional datasets $\mathcal{D}^{\text{obs}} \in \mathbb{R}^{n_{\text{obs}} \times K}$, $\mathcal{D}^{\text{int}} \in \mathbb{R}^{n_{\text{int}} \times K}$ are encoded into tensors $\mathbf{H}^{\text{obs}}, \mathbf{H}^{\text{int}} \in \mathbb{R}^{n_{(\cdot)} \times K \times d}$ whose feature dimension $d = d_v + d_n + d_t + d_a$ is partitioned into four *concatenated* segments. Each scalar $x_{s,k}$ is projected by a data-type-specific linear layer and concatenated with a node-identity, a sample-type, and an anomaly segment:

$$\mathbf{H}^{\text{obs}}_{s,k} = \big[\, W^{\text{obs}} x_{s,k} \,\|\, \mathbf{e}^{\text{node}}_k \,\|\, \mathbf{e}^{\text{type}}_0 \,\|\, \mathbf{e}^{\text{anom}}_{a_k} \,\big], \qquad \mathbf{H}^{\text{int}}_{s,k} = \big[\, W^{\text{int}} x_{s,k} \,\|\, \mathbf{e}^{\text{node}}_k \,\|\, \mathbf{e}^{\text{type}}_1 \,\|\, \mathbf{e}^{\text{anom}}_{a_k} \,\big], \tag{E.1}$$

where $W^{\text{obs}}, W^{\text{int}} \in \mathbb{R}^{d_v \times 1}$ are separate learned value embeddings. Since observational and interventional data are different distributions, each gets its own weights rather than a shared projection. $\mathbf{e}^{\text{node}}_k \in \mathbb{R}^{d_n}$ is a *fixed*, non-learned sinusoidal positional embedding over node index (Vaswani et al., 2017). We do since node order is randomly permuted and carries no intrinsic structure. $\mathbf{e}^{\text{type}}_c \in \mathbb{R}^{d_t}$ $c \in \{0, 1, 2\}$ are the observational and interventional/queries respectively and $\mathbf{e}^{\text{anom}}_a \in \mathbb{R}^{d_a}$ ($a \in \{0, 1, 2\}$ is the non-target/target/mask-not-provided $a_k = m_k$ when the binary symptom mask $m \in \{0, 1\}^K$ is available, else $a_k = 2$.

A learned per-node query token $\mathbf{z}^{(0)}_k = \big[\, \mathbf{q}_k \,\|\, \mathbf{e}^{\text{type}}_2 \,\|\, \mathbf{e}^{\text{anom}}_{a_k} \,\big]$, with free parameter $\mathbf{q}_k \in \mathbb{R}^{d_v + d_n}$, is concatenated onto the sample axis as one additional value giving $\mathbf{Z}^{(0)} = [\mathbf{H}^{\text{obs}}; \mathbf{H}^{\text{int}}; \mathbf{z}^{(0)}] \in \mathbb{R}^{N \times K \times d}$ with $N = n_{\text{obs}} + n_{\text{int}} + 1$.

**Alternating-Attention Transformer Blocks** $L$ pre-norm transformer blocks refine $\mathbf{Z}^{(0)}$ as a single sequence: observational and interventional samples attend to each other at every layer. (1) node-level attention (even $\ell$): for each of the $N$ samples, attention is computed across the $K$ node positions; (2) sample-level attention (odd $\ell$): for each node, attention is computed jointly across all $N$ samples – observational, interventional, and the query token. Each attention and each MLP sublayer sits in a pre-norm residual block:

$$\hat{\mathbf{Z}}^{(\ell)} = \mathbf{Z}^{(\ell-1)} + \text{Attn}^{(\ell)}\big(\text{LN}(\mathbf{Z}^{(\ell-1)})\big), \qquad \mathbf{Z}^{(\ell)}_{\text{pre}} = \hat{\mathbf{Z}}^{(\ell)} + \text{MLP}\big(\text{LN}(\hat{\mathbf{Z}}^{(\ell)})\big), \tag{E.2}$$

where "+" denotes the residual (skip) connection. After each block, the type and anomaly segments are *re-written* . This means they are overwritten, not added to with $\mathbf{e}^{\text{type}}, \mathbf{e}^{\text{anom}}$, so every token's identity stays valid. However deep the stack is; the value/node segments are left to evolve through the residual stream:

$$\mathbf{Z}^{(\ell)} = \big[\, \mathbf{Z}^{(\ell)}_{\text{pre},\, 1:d_v+d_n} \,\|\, \mathbf{e}^{\text{type}} \,\|\, \mathbf{e}^{\text{anom}} \,\big]. \tag{E.3}$$

**Decoding** After $L$ layers a final layer norm is applied, $\bar{\mathbf{Z}} = \text{LN}_f(\mathbf{Z}^{(L)})$, and the sequence is split into the embedded samples $\bar{\mathbf{H}} = \bar{\mathbf{Z}}_{1:n_{\text{obs}}+n_{\text{int}}}$ and the embedded query token $\bar{\mathbf{z}} = \bar{\mathbf{Z}}_N$. Another attention readout (QueryPooling) maps the dataset directly to a summary/embedding for each node

$$\mathbf{u} = \text{LN}\Big(\bar{\mathbf{z}} + \text{CrossAttn}\big(\bar{\mathbf{z}}, \bar{\mathbf{H}}, \bar{\mathbf{H}}\big)\Big) \in \mathbb{R}^{K \times d}. \tag{E.4}$$

As no absolute positional encoding is used along the sample axis, the alternating self-attention layers and the final cross-attention layer are both invariant to the ordering of the observational and interventional samples. This means that the difference between them is instead learned from the type segments rather than imposed as an explicit difference. A two-layer MLP decoder then maps each node's summary to a logit: $\hat{T} = c_\psi(\mathbf{u}) \in \mathbb{R}^K$.

# F Hyperparameters and Compute

We train a total of two models as some of the smaller experiments can be done with a simpler/smaller model trained with a smaller maximum number of variables size $K_{max}$. All models use embedding dimension $d = 160$, $\updownarrow = 6$ attention layers, 8 attention heads, MLP hidden size 512, and dropout 0.1. The small model is trained on 80,000 unique samples from 20,000 distinct SCMs; the large model on 7,680,000 samples from 1,920,000 distinct SCMs. The small model is trained with a single root cause per episode ($|T| = 1$); the large model is trained on the prior's default multi-root-cause mix, where the number of root causes is uniformly sampled between 1 and 3(Appendix D.2), so that it also covers the multiple simultaneous root causes present in the CausRCA benchmark. Hyperparameters for all training runs and experimenst are listed in Table F.1; compute requirements are summarised in Table F.3; the model used per experiment is given in Table F.2.

Table F.1: Hyperparameters for training and experiments

| Component | Hyperparameter | Value |
|---|---|---|
| Training (all models) | Optimizer | AdamW |
| | Learning rate | $5 \times 10^{-4}$ |
| | Weight decay | 0.01 |
| | Episodes / epoch | 1,000 |
| | Queries / episode ($n_q$) | 4 |
| | $n_{\text{obs}}$ | $\text{Unif}(5,\ 500)$ |
| | $n_{\text{int}}$ | $\text{Unif}(1,\ 200)$ |
| | Dropout | 0.1 |
| Training (per model) | Small ($K_{\min} = 2,\ K_{\max} = 5$), 40 epochs | 4×A100-80GB |
| | Large ($K_{\min} = 17,\ K_{\max} = 100$), 160 epochs | 8×A100-80GB |
| Multi-RC FT | Learning rate | $3 \times 10^{-5}$ |
| | Epochs | 5 |
| | $n_{\text{obs}}$ | $\text{Unif}(40,\ 150)$ |
| | $n_{\text{int}}$ | $\text{Unif}(5,\ 20)$ |
| Ablation FT | Learning rate | $5 \times 10^{-5}$ |
| | Epochs | 10 |
| | Episodes / epoch | 500 |

Table F.2: Model used per experiment.

| Experiment | Model | $K_{\max}$ |
|---|---|---|
| Three-node confounder / mediator | PRIM-S | 10 |
| Finetuning ablation (sinusoidal, far OOD) | PRIM-S | 10 |
| Finetuning ablation (NN, near OOD) | PRIM-S | 10 |
| Multi-root-cause identification | PRIM-S | 10 |
| Scalability (20–100 nodes) | PRIM-L | 100 |
| PetShop | PRIM-L | 100 |
| CausRCA | PRIM-L | 100 |
| Inference time benchmark | PRIM-L | 100 |

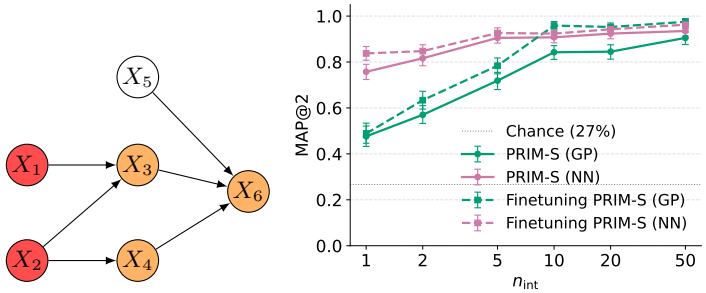
Table F.3: Compute requirements. Epochs serve solely as a logging mechanism.

| Component | Parameters | Hardware | Wall time | GPU-hrs |
|---|---|---|---|---|
| Small model | 1,928,409 | 4×A100-80GB | 41m | $\approx 5.5$ |
| Large model | 1,939,209 | 8×A100-80GB | 76h 37m | $\approx 612.9$ |
| Evaluation | — | 1×A100 | 2h | $\approx 2.0$ |
| **Total (reported)** | | | | $\approx 620$ |
| **Total (incl. trials)** | | | | $\approx 3102$ |

# G Additional Experiments

## G.1 Multi-Root Cause Identification

Here PRIM's ability to identify multiple simultaneous root causes on the fixed DAG shown in Figure G.1 is tested, where two independent nodes $X_1$ and $X_2$ are intervened upon simultaneously. Performance is measured by MAP@2, requiring both root causes to appear in the top-2 predictions, testing zero-shot transfer from single to multi-fault identification. The fine-tuned variant is trained on 1,600 synthetic multi-fault scenarios with mixed NN/GP mechanisms and full-model fine-tuning; details are given in Appendix F.




(a) Varying $n_{\text{int}}$, keeping $n_{\text{obs}} = 100$

Figure G.1: Multi-root-cause evaluation on a 6-node DAG (left). $X_1, X_2$ (■ red) are root causes; $X_3, X_4, X_6$ (■ orange) are descendants; $X_5$ (white) is unaffected. MAP@2 measures whether both appear in the top-2 predictions. Solid: zero-shot; dashed: finetuned on multi-fault DAGs.

As shown in Figure G.1 for a fixed $n_{\text{obs}} = 100$, the fine-tuned variant substantially outperforms the base model, with gains becoming more pronounced as $n_{\text{int}}$ increases and GP mechanisms are handled more reliably. Notably, even the base model exceeds the random MAP@2 baseline of $\approx 0.27$ with sufficient interventional samples, suggesting that single-fault training transfers partially to the multi-fault setting.

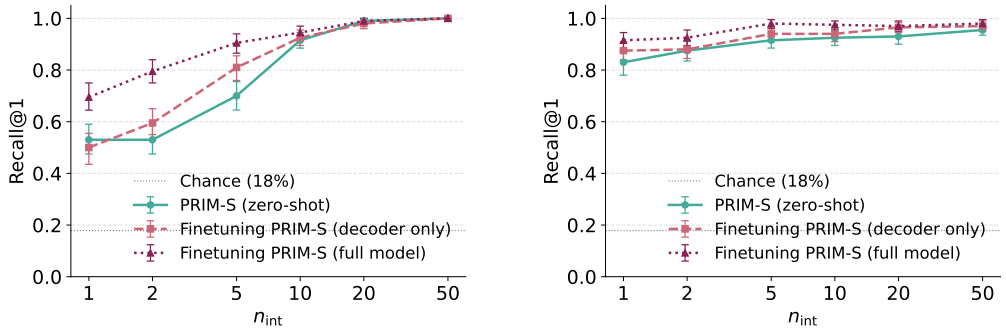
### G.2 Finetuning and out of Distribution Generalisation

We evaluate the benefit of finetuning and the choice of finetuning variants across two out-of-distribution settings. This also showcases how PRIM adapts to out of distribution data.

In both cases we use random DAGs with 3–10 nodes and sweep over the number of interventional samples $n_{\text{int}}$, holding $n_{\text{obs}} = 100$ fixed. In both settings finetuning uses AdamW with learning rate $5 \times 10^{-5}$, weight decay 0.01, 10 epochs of 500 episodes each, and $n_{\text{obs}} = 100$ fixed. They both use $k_{max} = 10$

*Sinusoidal* (far OOD): structural equations of the form $f(x) = a\sin(bx + c)$ with randomly sampled parameters, entirely absent from the meta-training prior, and an intervention strength of $2\times$ (vs. $3\times$ during training). *NN equations* (near OOD): the same NN equation family as training but with a reduced intervention strength of $1.5\times$ (vs. $3\times$ during training), weakening the $\mathbf{\Delta}$ signal while keeping the functional form in-distribution. Three conditions are compared: zero-shot (no finetuning), decoder-only finetuning (backbone frozen), and full-model finetuning (all parameters updated). Finetuning in both cases uses synthetic data drawn from the respective target distribution.

As shown in Figure G.2, finetuning consistently reduces the number of interventional samples needed to achieve high recall, with the largest gains at small $n_{\text{int}}$. For the far-OOD sinusoidal setting (Figure G.2a), all three conditions are clearly separated: full-model finetuning achieves $\approx$0.70 recall at $n_{\text{int}} = 1$ versus $\approx$0.45 for zero-shot, with decoder-only falling between the two. This indicates that the backbone itself needs to adapt to the novel functional form, and updating all parameters is beneficial. For the near-OOD NN setting (Figure G.2b), zero-shot performance is already high ($\approx$0.83 at $n_{\text{int}} = 1$) despite the reduced intervention strength, and decoder-only finetuning nearly matches full-model finetuning across the entire sweep. This is consistent with the backbone already producing well-calibrated embeddings for the in-distribution functional form. However, with sufficient interventional data PRIM also converges to the true root causes even when the posterior is out of its prior distribution.




(a) Sinusoidal equations (far OOD)

(b) Weak-signal NN equations (near OOD)

Figure G.2: Recall@1 vs. $n_{\text{int}}$ at fixed $n_{\text{obs}} = 100$ for three finetuning conditions. Finetuning uses synthetic data from the target distribution. Error Bars show 90% bootstrap confidence intervals.

## G.3 Additional Results Different Node Sizes

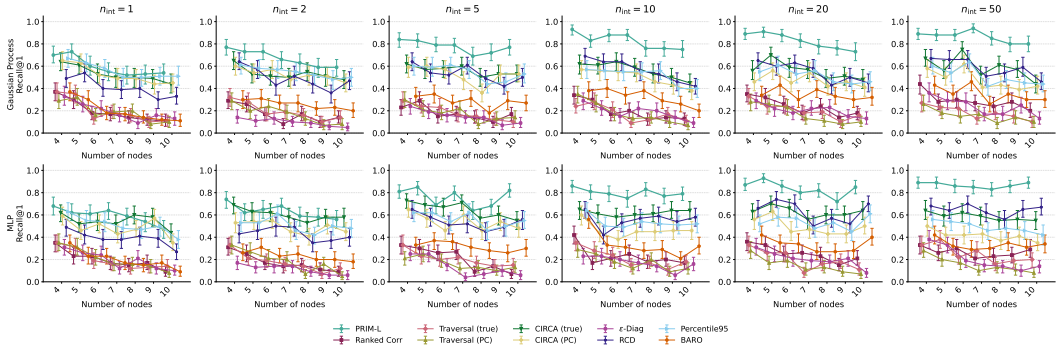



Figure G.3: Recall@1 vs. number of nodes across all observation sizes ($n_{\text{obs}} \in \{50, 100, 200, 500\}$, $n_{\text{int}} = 10$). Rows correspond to data generating process (Gaussian Process and Neural Network); columns correspond to number of observations. For smaller Model (3 to 10 Nodes)

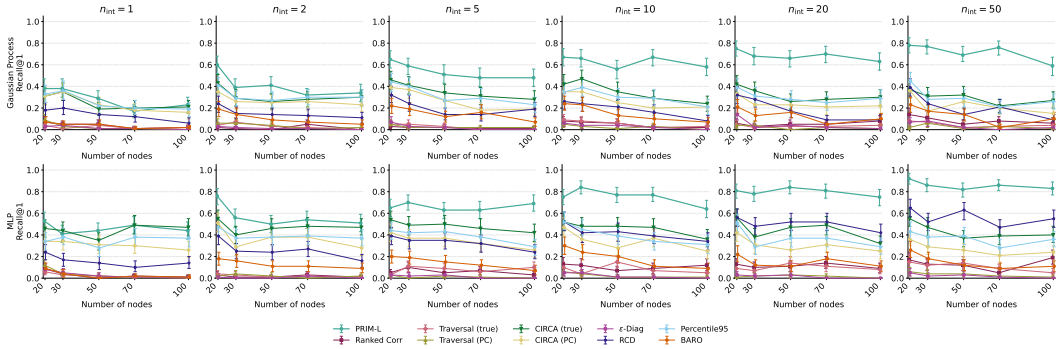



Figure G.4: Recall@1 vs. number of nodes across all observation sizes ($n_{\text{obs}} \in \{50, 100, 200, 500\}$, $n_{\text{int}} = 10$). Rows correspond to data generating process (Gaussian Process and Neural Network); columns correspond to number of observations. For large Model trained on (17 - 100 Nodes)

### G.4 Inference Time

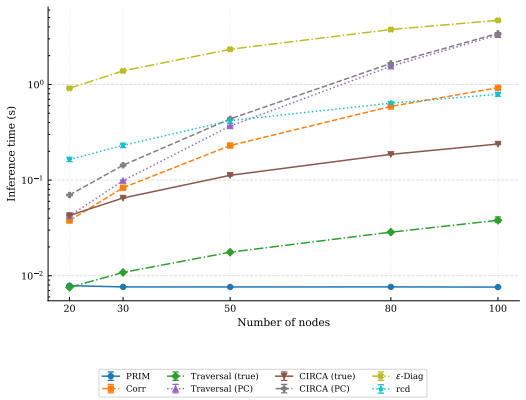



Figure G.5: Inference time for different node sizes for different methods $n_{\text{obs}} = 100$, $n_{\text{int}} = 20$

Table G.1: PRIM inference time on a MacBook. Mean and 90% bootstrap CI over 100 episodes per configuration ($n_{\text{obs}} = 100$, $n_{\text{int}} = 20$).

| Device | $K$ | Mean (ms) | $\text{CI}_{\text{lo}}$ (ms) | $\text{CI}_{\text{hi}}$ (ms) |
|---|---|---|---|---|
| CPU | 20 | 115.8 | 115.2 | 116.5 |
| | 30 | 117.2 | 116.8 | 117.6 |
| | 50 | 119.7 | 119.3 | 120.2 |
| | 80 | 122.6 | 122.2 | 123.0 |
| | 100 | 124.5 | 124.1 | 125.0 |
| MPS (Apple GPU) | 20 | 17.4 | 16.7 | 18.5 |
| | 30 | 16.8 | 16.8 | 16.9 |
| | 50 | 16.6 | 16.6 | 16.7 |
| | 80 | 16.5 | 16.5 | 16.6 |
| | 100 | 16.5 | 16.4 | 16.5 |

### G.5 Additional Results For Petshop

Table G.2: Top-1 recall of the RCA methods measuring the accuracy of identifying the correct root-cause node. Results marked with * are taken from Hardt et al. (2024); † denotes our method. **Bold** indicates the best result per row overall; blue indicates the best result per row among graph-not-given methods.

| | | graph given | | | graph not given | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| traffic scenario | metric | traversal* | circa* | counter-factual* | $\varepsilon$-diag.* | rcd* | corr* | Baro | TimeRank | PRIM† |
| low | latency | **0.57** | 0.36 | 0.36 | 0.00 | 0.07 | 0.43 | 0.00 | 0.00 | 0.30 |
| low | availability | 0.50 | 0.42 | 0.00 | 0.00 | 0.58 | **0.75** | 0.00 | 0.00 | 0.63 |
| high | latency | 0.57 | 0.50 | 0.57 | 0.00 | 0.00 | **0.64** | 0.00 | 0.00 | 0.50 |
| high | availability | 0.33 | 0.00 | 0.00 | 0.00 | 0.00 | **0.83** | 0.00 | 0.00 | 0.38 |
| temporal | latency | **1.00** | 0.75 | 0.38 | 0.12 | 0.25 | 0.62 | 0.00 | 0.00 | 0.33 |
| temporal | availability | 0.38 | 0.38 | 0.00 | 0.00 | 0.50 | 0.62 | 0.00 | 0.00 | **0.67** |
| average | | 0.56 | 0.40 | 0.22 | 0.02 | 0.23 | **0.65** | 0.00 | 0.00 | 0.46 |

Table G.3: Top-3 recall of the RCA methods for the PetShop data measuring the accuracy of including the correct root-cause node in the top-3 results. Results marked with * are taken from Hardt et al. (2024); † denotes our method. **Bold** indicates the best result per row overall; blue indicates the best result per row among graph-not-given methods.

| | | graph given | | | graph not given | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| traffic scenario | metric | traversal* | circa* | counter-factual* | $\varepsilon$-diag.* | rcd* | corr* | Baro | TimeRank | PRIM† |
| low | latency | 0.57 | **0.86** | 0.71 | 0.00 | 0.21 | 0.57 | 0.30 | 0.10 | 0.80 |
| low | availability | **1.00** | **1.00** | 0.42 | 0.00 | 0.75 | 0.92 | 0.00 | 0.00 | 0.88 |
| high | latency | 0.79 | **1.00** | 0.86 | 0.00 | 0.07 | 0.79 | 0.00 | 0.30 | 0.60 |
| high | availability | **1.00** | 0.00 | 0.00 | 0.33 | 0.00 | 0.92 | 0.00 | 0.00 | 0.88 |
| temporal | latency | **1.00** | **1.00** | 0.50 | 0.12 | 0.75 | 0.75 | 0.17 | 0.17 | 0.67 |
| temporal | availability | **1.00** | **1.00** | 0.25 | 0.12 | 0.75 | 0.75 | 0.00 | 0.33 | **1.00** |
| average | | **0.86** | 0.81 | 0.46 | 0.10 | 0.39 | 0.78 | 0.08 | 0.15 | 0.79 |

### G.6 Additional Results for CausRCA

Table G.4: MAP@3 for RCA algorithms on the CausRCA benchmark. Results marked with * are taken from Mehling et al. (2026); †Ours. **Bold** indicates best result per row overall; blue indicates best result per row among graph-not-given methods.

| Dataset | | graph given | | | | | graph not given | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | CausTR* | Rand Walk* | Page Rank* | Traversal | CIRCA | Time Rank* | Baro* | Corr | CausTR +PC* | $\varepsilon$-Diag | RCD | PRIM† |
| Coolant | Full | **0.99** | 0.00 | 0.00 | 0.00 | 0.20 | 0.35 | 0.00 | 0.00 | – | 0.00 | 0.01 | 0.457 |
| | Sub | **1.00** | 0.14 | **1.00** | 0.00 | 0.21 | 0.85 | 0.88 | 0.15 | 0.87 | 0.00 | 0.41 | 0.760 |
| Hydraulics | Full | **0.74** | 0.00 | 0.11 | 0.00 | 0.17 | 0.18 | 0.26 | 0.13 | – | 0.00 | 0.08 | 0.313 |
| | Sub | 0.74 | 0.14 | **0.98** | 0.00 | 0.20 | 0.47 | 0.75 | 0.35 | 0.48 | 0.02 | 0.21 | 0.745 |
| Probe | Full | **0.57** | 0.00 | 0.00 | 0.00 | 0.01 | 0.19 | 0.00 | 0.00 | – | 0.06 | 0.01 | 0.086 |
| | Sub | 0.58 | 0.32 | 0.26 | 0.00 | 0.23 | 0.47 | **0.63** | 0.45 | 0.37 | 0.29 | 0.14 | 0.353 |
| average | Full | **0.77** | 0.00 | 0.04 | 0.00 | 0.12 | 0.24 | 0.09 | 0.05 | – | 0.02 | 0.04 | 0.272 |
| | Sub | **0.78** | 0.20 | 0.75 | 0.00 | 0.21 | 0.60 | 0.75 | 0.33 | 0.57 | 0.11 | 0.24 | 0.616 |

## NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims stated in introduction and abstract are supported by empirical data and are referred to later in the conclusion

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitations are stated in conclusion. The assumptions made are clearly stated especially for the identifiability of the posterior when it is introduced.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.

- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: We derive the Posterior in the appendix and state all the assumptions needed to apply the proof of Squires et al. (2020) to our setting

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: All the experiments can be reproduced by the methods given in the main body and the additional appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case

of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: We submit the code with the submission as a zip file and release the code publicly on acceptance.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: All the main Hyperparameter are either listed in the paper itself or the appendix the hyperparamneter for the model are in appendix F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Provide error bars over bootstrap intervals for synthetic data, do not report this for PetShop and CausRCA as it is classification and the mean actually gives enough information about the confidence. This is in line of how the results were reported by Hardt et al. (2024) and Mehling et al. (2026) respectively.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The compute needed is clearly stated in appendix F.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

Justification: We reviewed the code of ethics and the research conducted is in line with it.

Guidelines:

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

10. **Broader impacts**

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

Answer: [N/A]

Justification: As RCA is a niche problem with main impacts in industry we do not expect societal impact.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A]

Justification: As for the societal impact we do not expect any dual use scenarios.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: CausRCA and Petshop data properly cited and licence mentioned all methods clearly referenced.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: Creation for the code and data is clearly mentioned.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A]

    Justification: No crowdsourcing

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [N/A]

    Justification: No research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLM only used for code formatting and help of writing not fro development of core methods.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.