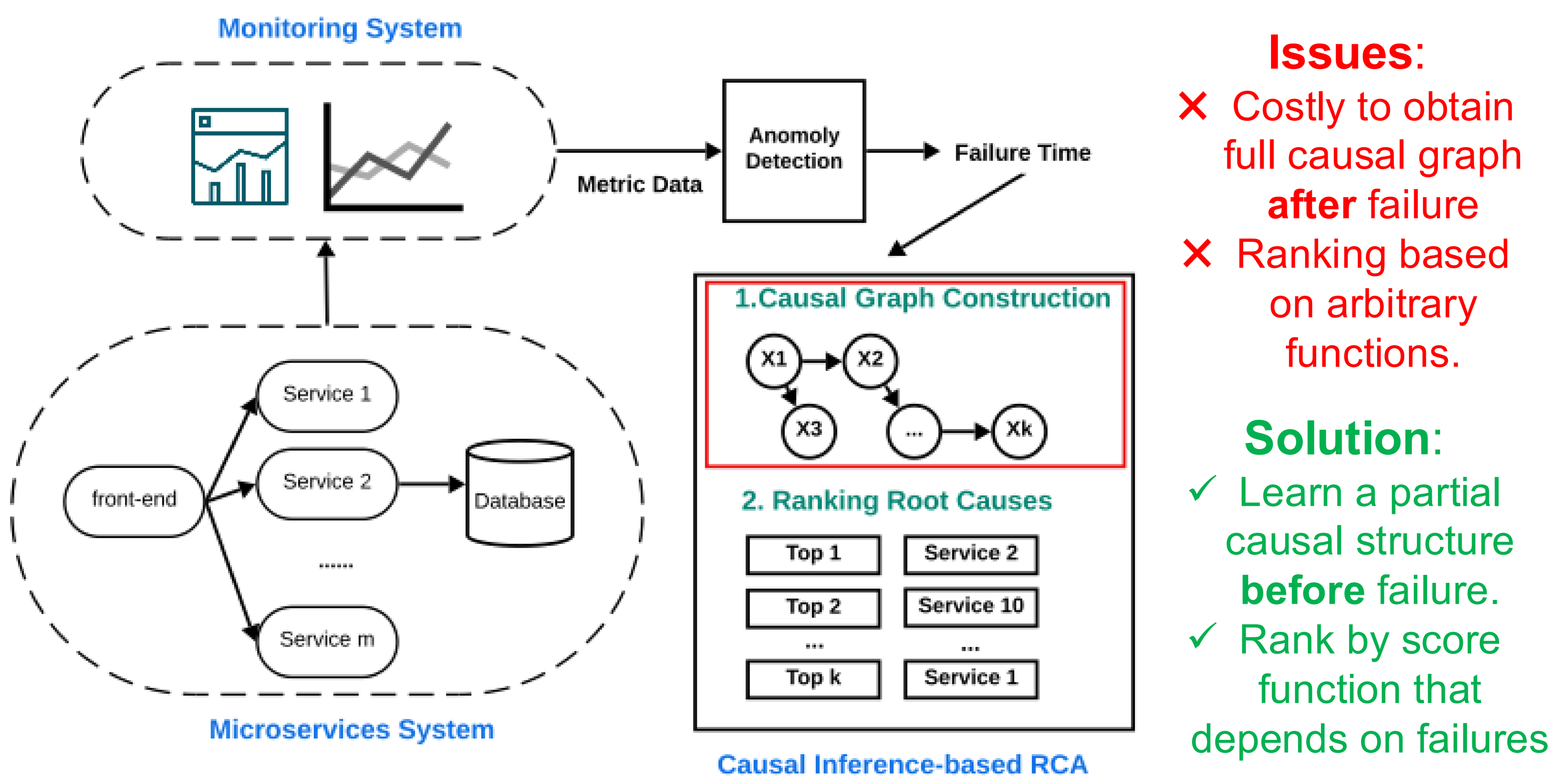
Root Cause Analysis of Failure from Partial Causal Structures

Azam Ikram, Kenneth Lee, Shubham Agarwal, Shiv Kumar Saini, Saurabh Bagchi, Murat Kocaoglu

Purdue University, Adobe Research

Paper Code

What is Root Cause Analysis?



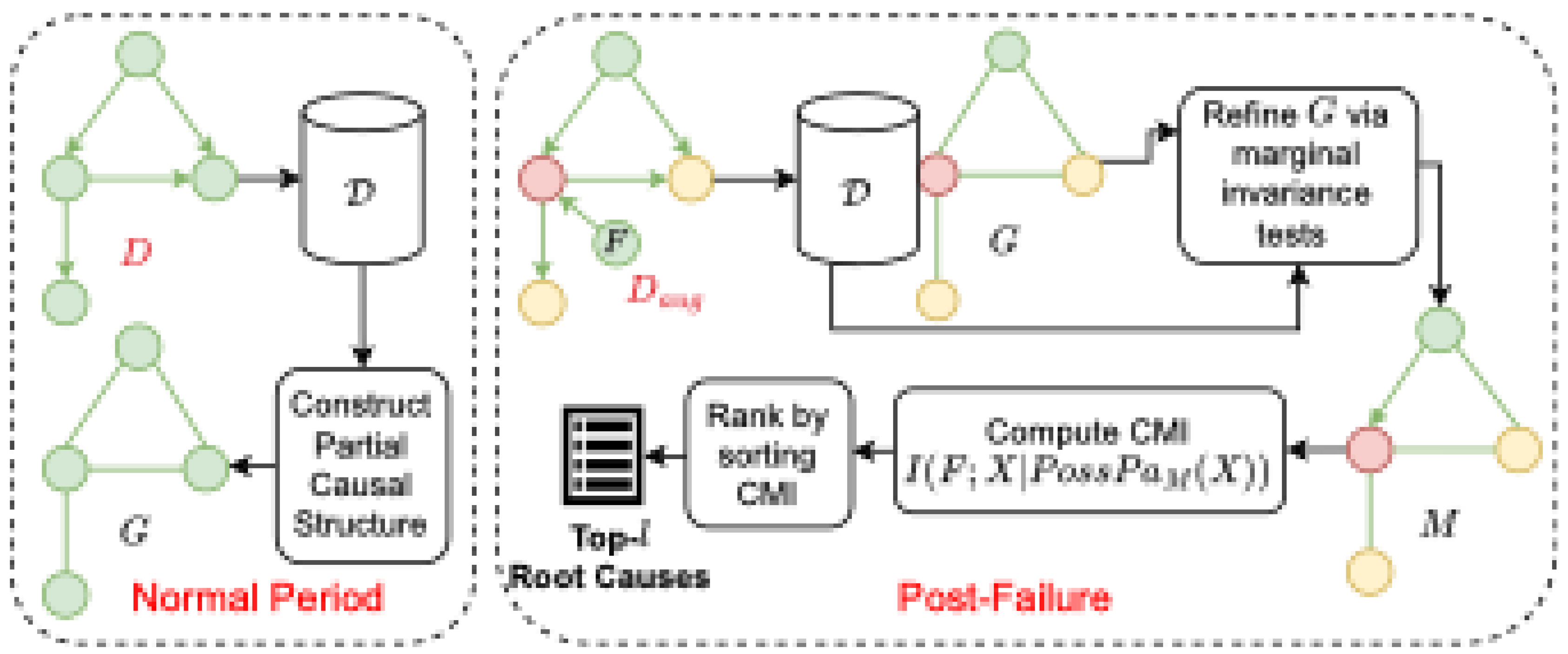
Issues:

- ✕ Costly to obtain full causal graph **after** failure
- ✕ Ranking based on arbitrary functions.

Solution:

- ✓ Learn a partial causal structure **before** failure.
- ✓ Rank by score function that depends on failures

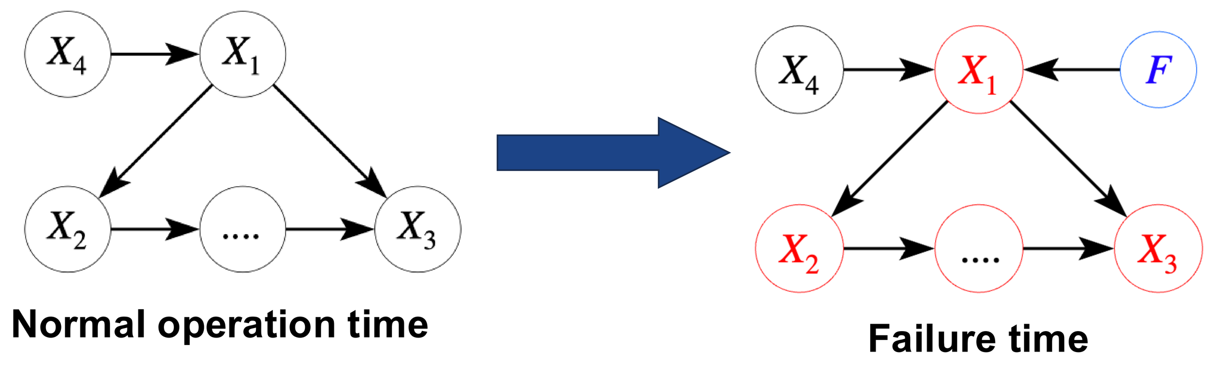
Proposed Algorithm (RCG)



1. Learn a **partial causal structure** e.g. CPDAG/k-essential graphs during **normal operation time.**
2. Apply **n marginal invariance tests.**
3. Compute **conditional mutual information (CMI) e.g., I(F;X|PossPa(X)).**
4. Rank all variables from highest CMI to the lowest to find top-k root causes.

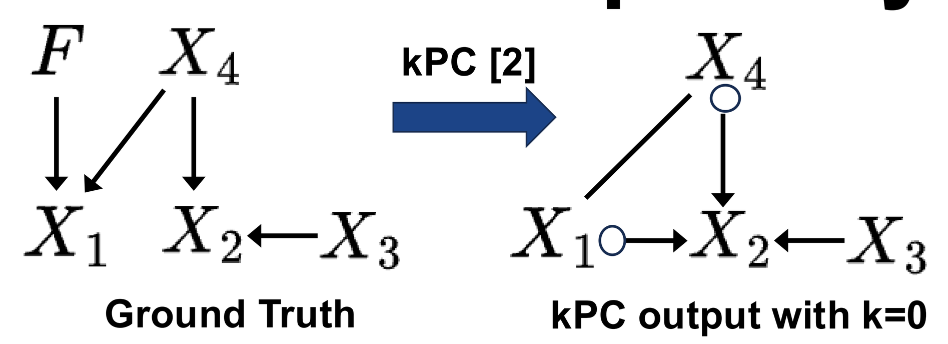
Theorem: RCG algorithm returns the true root cause variables under causal sufficiency and the extended faithfulness assumptions.

Modeling Failures as Interventions



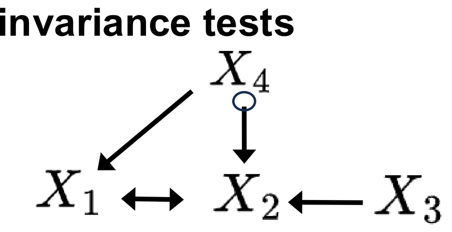
Before and after a failure can be modeled as a *soft* intervention indicated by a binary variable F [1].

RCG adopts beyond CPDAG



RCG

1. Apply marginal invariance tests



2. Compute CMI between F and each variable given their possible parents

$I(F; X_4)$ $I(F; X_3)$

$I(F; X_1|X_4)$

$I(F; X_2|X_3, X_4)$

Key idea

Conditioning on Possible Parents 'blocks' the paths from F to X unless X is a root cause.

How to find unknown targets with marginal tests ONLY?

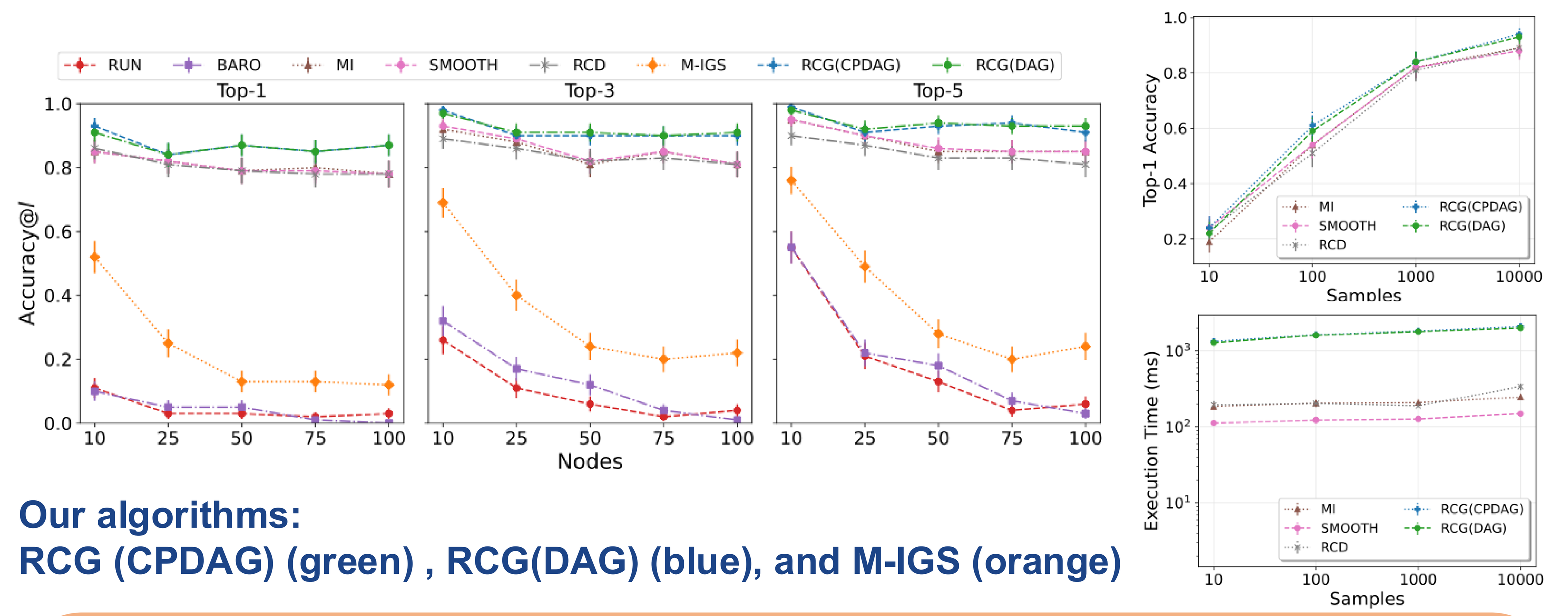
Finding a single root cause given a DAG with marginal tests only

IGS: Given an oracle to answer Y/N whether there is a directed path to R, what is the minimum number of queries to identify R in a DAG?

Theorem: Given a DAG, any algorithm the **only uses marginal invariance tests must perform** $\Omega(\log_2 n + d\log_{1+d} n)$ many tests to find the single root cause in the worst case, where d is the maximum in-degree of D and n is the number of nodes. There exists an algorithm that finds the root cause with $O(\log_2 n + d\log_{1+d} n)$ marginal invariance tests.

Experimental Results

Case 1: Perfect CPDAG from observed data; 100 random DAGs with degree 2 and 4 states with 1000 failure samples



Our algorithms:
RCG (CPDAG) (green) , RCG(DAG) (blue), and M-IGS (orange)

Accuracy:

- Although **M-IGS** offers the lowest runtime among all CI-based methods, its accuracy drops sharply. This decline stems from a key limitation: **M-IGS** assumes perfect CI tests, but in practice, test results can be unreliable due to limited sample availability.
- **RCG** outperforms baselines due to its reliance on the correct CPDAG and its ability to orient edges accurate after failure.

Efficiency:

- **RCG** also is more efficient **as it only need to conduct n marginal tests and CMI computations.**

Case 2: A real-world production application from Jan to Jul 2024, during which four outages were reported. For each incident, Software Reliability Engineers documented key details, including outage duration, detection time, resolution method, root cause.

Outage	RCG-0	RCG-1	RCG-2	MI	BARO	RCD
A	7	-	-	-	9	-
B	1	6	-	9	6	-
C	1	1	1	1	8	1
D	5	5	6	3	-	2

Rank of the true root cause given by each algorithm

Outage	Nodes	Normal Samples	Failure Samples	Duration (hours)
A	152	4783	918	15
B	141	4626	1217	20
C	149	3464	110	2
D	146	7165	567	5

Statistics of the dataset

Applicability to real-world use case:

- After incorporating **k-PC** algorithm [2], **RCG** outperforms most baselines in all four outages even when the failure samples are limited.

Reference

[1] Ikram, Azam, et al. "Root cause analysis of failures in microservices through causal discovery." Advances in Neural Information Processing Systems 35 (2022): 31158-31170.

[2] Kocaoglu, Murat. "Characterization and learning of causal graphs with small conditioning sets." Advances in Neural Information Processing Systems 36 (2023): 74140-74179.