LEARNING APPLICABILITY ENVELOPES FOR CONDITIONAL INDEPENDENCE TESTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Conditional independence (CI) testing is a foundational task in statistical inference, yet even for discrete data, classical CI-test methods fail in different and often incompatible regimes. This raises a practical question: *given a specific CI-testing instance, which test method should be trusted?* To study this question, we first synthesize verifiable data and post-train a large language model to combine the outputs of four representative CI-test methods, χ^2, Fisher's exact, mutual information (MI), and logistic regression, into a final CI decision, based on the outputs of the CI tests and a four-dimensional feature vector characterizing the difficulty of the instance through observable data-quality and conditioning properties. We then analyze the router's learned policy in method-disagreement regimes and extract quantitative decision thresholds over the feature space to analyze **applicability envelopes** of the different CI-test methods. We validate these thresholds against reference transitions in test error rates and examine their variation across data regimes. Overall, our LLM router achieves the highest F1 on our synthetic evaluation set and shows competitive transfer performance with matched non-LLM routers on three additional Bayesian networks and the Sachs dataset. Furthermore, our analyses identify boundaries consistent with changes in test reliability and connect an MI boundary to the treatment of collapsed strata in the tested implementation's degrees of freedom. Finally, we evaluate the router on the Sachs flow-cytometry dataset at both the CI-testing level and the downstream causal-discovery level, where it achieves the best skeleton recovery among the evaluated PC variants. Our code is available at `https://anonymous.4open.science/r/CI-testing_router-0D00`.

1 INTRODUCTION

Conditional independence is a foundational concept in both theory and practice of statistical inference. In theory, it unifies many fundamental ideas from statistical inference such as sufficiency and ancillarity (Dawid, 1979). In practice, conditional independence testing (CI-testing) operationalizes the concept and is broadly used in many downstream frameworks like constraint-based causal discovery (Spirtes et al., 2000), graphical-model learning (Tsamardinos et al., 2006), fairness auditing (Hardt et al., 2016) and feature selection (Brown et al., 2012). Because of its wide-use, errors in the CI-testing step can have significant impact due to downstream error propagation.

Diverse CI-test methods have been developed and refined extensively, but none dominates another universally in practical settings. For example, for discrete random variables, there are four canonical families of tests: asymptotic tests such as those based on the χ^2 and G^2 statistics (Pearson, 1900; Wilks, 1938), information-theoretic tests such as mutual information-based tests (Panzeri & Treves, 1996), exact tests (Fisher, 1922; Boschloo, 1970), and regression-based tests (Agresti, 2013). Given their decades of refinement, one may expect convergence on a particular test that works well and dominates all others. However, we show evidence to the contrary in our preliminary study: our analysis found that the different methods respond to test difficulties differently, many of which arise from data-quality and conditioning properties (Cochran, 1954; Agresti, 2013). In particular, high conditioning order fragments samples across exponentially many strata in contingency tables, a discrete manifestation of the curse of dimensionality (Bellman, 1954). Finite sample sizes further induce sparsity, where some cells are empty, and degeneracy, where some rows or columns collapse.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

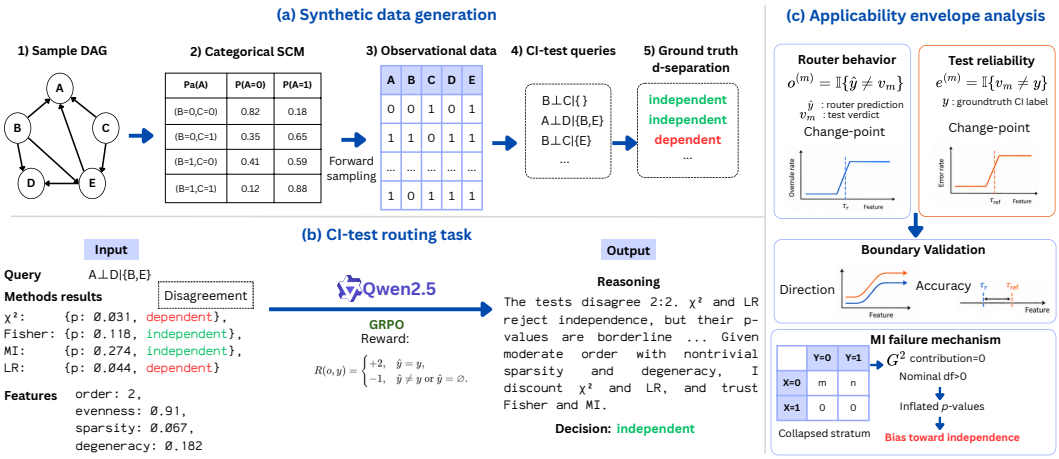


Figure 1: Overview of synthetic CI-instance generation, verifiable router training, and applicability-envelope analysis. Candidate router boundaries are compared with reference test-error transitions, and an MI boundary motivates a diagnostic of the tested implementation.

This naturally leads to the notion of an **applicability envelope**: the boundaries of a test method's reliability across different data conditions. We focus on problems where a group of CI tests disagree, since these cases can reveal differences in the methods' applicability envelopes and are where such envelopes can guide method selection in practice.

Existing guidance about such envelopes remains largely qualitative: χ^2 tests become unreliable when expected counts are too small, Fisher's exact tests are preferred at small sample sizes, and mutual-information estimators are biased in sparse tables (Panzeri & Treves, 1996). Even numerical criteria such as Cochran's expected-count guidelines (Cochran, 1954) are better understood as rules of thumb than as calibrated decision boundaries. We therefore seek quantitative applicability envelopes: regime-dependent boundaries that characterize changes in test reliability and inform which evidence to trust. This leads to our research question: *Can we learn quantitative, instance-level applicability envelopes for classical CI tests, so as to decide which test to trust on a given finite-sample problem?*

As shown in Figure 1, we formulate quantitative-envelope discovery as a learning problem and post-train a reasoning large language model (LLM) to make CI decisions from a CI-testing query, instance-level data-quality and conditioning-difficulty features, and the outputs of four classical CI tests. We use the reinforcement-learning-with-verifiable-rewards (RLVR) paradigm (Wang et al., 2023; Guo et al., 2025), which is well suited to our setting because each router decision can be checked against a verifiable CI label. To construct such verifiable training instances, we sample synthetic causal graphical models, assign ground-truth CI labels by D-separation (Geiger et al., 1990) under the standard faithfulness assumption (Spirtes et al., 2000; Pearl, 2009; Meek, 2013), and forward-sample discrete observational data to construct CI-testing instances.

On held-out synthetic instances, the RLVR router achieves the highest F1 among the evaluated methods. Comparisons with five matched non-LLM routers further show competitive transfer performance across three expert-built Bayesian networks and the Sachs flow-cytometry dataset (Sachs et al., 2005), without target-specific retraining. Within PC skeleton discovery (Spirtes et al., 2000) on the Sachs dataset, the RLVR router achieves the lowest skeleton Hamming distance and highest skeleton F1 among the evaluated methods.

Analysis of the learned routing policy provides empirical evidence for regime-dependent applicability envelopes. Across five datasets, RLVR achieves the highest boundary-recovery precision among the evaluated routers, while recovering a subset of reference-supported transitions. Sample-size and cardinality analyses show that the numerical thresholds change with the data regime, while the fixed router continues to identify transitions in test reliability. Following the router's MI boundary leads to an implementation-level explanation: collapsed conditioning strata contribute zero to the tested MI statistic while still adding nominal degrees of freedom, inflating p-values. Recomputing

per-stratum effective degrees of freedom improves MI accuracy on the diagnostic pool, linking the learned boundary to a concrete source of test error.

We summarize our contributions as follows. **(I)** Verifiable CI-routing formulation: We formulate discrete CI-test selection as a verifiable learning problem by generating synthetic CI-testing instances from causal graphical models, assigning ground-truth labels by D-separation, and post-training an RLVR-style LLM router to resolve disagreements among classical CI tests. **(II)** Quantitative applicability-envelope analysis: We develop a model-agnostic procedure to extract candidate applicability boundaries from routing policies and validate their detection, direction, and localization against reference test-error transitions. This analysis connects learned routing behavior to test reliability and leads to an actionable diagnosis of a degrees-of-freedom issue in the tested MI implementation. **(III)** Empirical characterization of cross-regime transfer: Through matched comparisons with five non-LLM routers, we characterize predictive transfer and boundary recovery across synthetic data, three expert-curated Bayesian networks, and real-world Sachs measurements. Sample-size and cardinality analyses reveal regime-dependent reliability thresholds that the fixed router can track without retraining, while Sachs PC experiments assess the downstream benefits of CI routing.

2 Related Work

Conditional independence testing and applicability. Conditional independence (CI) testing has a long history in statistics and machine learning. Classical discrete tests include χ^2 test (Pearson, 1900), likelihood-ratio tests and their large-sample χ^2 approximation (Wilks, 1938), exact conditional tests rooted in Fisher's contingency-table analysis (Fisher, 1922) and later refinements such as Boschloo's test (Boschloo, 1970), regression or log-linear formulations, and information-theoretic tests based on conditional mutual information and its finite-sample bias (Panzeri & Treves, 1996). Practical use of these tests is shaped by rules such as Cochran's expected-count guidance for χ^2 approximations (Cochran, 1954). More recent work develops machine-learning-based CI-tests for continuous, nonlinear, and high-dimensional settings, including kernel-based conditional independence tests (Zhang et al., 2012), nearest-neighbor CMI tests (Runge, 2018), classifier-based approaches (Sen et al., 2017), and supervised causal-learning systems such as ML4C (Dai et al., 2023). These methods expand the testing toolbox, but they do not remove the central difficulty: no single procedure is uniformly reliable across sample sizes, conditioning dimensions, sparsity patterns, and modeling assumptions. Indeed, Shah & Peters (2020) show that CI-testing is intrinsically hard and argue that test suitability must be judged relative to the problem setting. Our work takes this observation as its starting point. Rather than proposing another CI-test, we study the applicability envelopes of existing test families: quantitative regions of problem space in which each method can be trusted, and the disagreement regimes in which such envelopes become visible.

LLM routing, tool use, and verifiable post-training. Our use of an LLM router is related to work on algorithm selection and tool-using language models. The classical algorithm-selection formulation asks how to choose, from a portfolio of methods, the algorithm best suited to a given problem instance (Rice, 1976; Smith-Miles, 2009). Recent LLM work studies a related capability in the form of tool use: models can be prompted or trained to decide when to call external tools, which tool to call, and how to incorporate the result into their reasoning (Yao et al., 2022; Schick et al., 2023; Patil et al., 2024). Separately, post-training with verifiable rewards has shown that language models can improve on reasoning tasks when correctness can be checked automatically rather than judged by preference (Lambert et al., 2024; Guo et al., 2025). Our setting lies at the intersection of these lines of work. The router combines test outputs and instance features, and its reward is mechanically verifiable. Unlike prior tool-use work, our goal is not merely to improve task accuracy: we use the trained router as an auditable object whose decisions can be distilled into quantitative applicability-envelopes.

3 Problem Formulation and Preliminary Study

3.1 Discrete Conditional Independence Testing

We consider conditional independence (CI) testing for discrete variables. Let X and Y be categorical target variables and let $Z = (Z_1, \ldots, Z_k)$ be a categorical conditioning set of order $k = |Z|$. Given

N i.i.d. observations $D = \{(X_i, Y_i, Z_i)\}_{i=1}^{N}$, a CI-test decides between

$$H_0 : X \perp Y \mid Z \qquad \text{and} \qquad H_1 : X \not\perp Y \mid Z.$$

We evaluate a method library $\mathcal{M} = \{\chi^2, \text{Fisher}, \text{MI}, \text{LR}\}$ covering four approaches to discrete CI testing. The χ^2 test compares observed and null-expected contingency-table counts using an asymptotic χ^2 reference distribution (Pearson, 1900). Fisher's exact test conditions on observed margins and evaluates association using hypergeometric reference distributions (Fisher, 1922). The information-theoretic method denoted as MI computes empirical conditional mutual information, implemented in the discrete case through the likelihood-ratio statistic $G^2 = \sum_z 2n_z \widehat{I}_z(X;Y)$ to measure departures from conditional independence, where n_z is the sample count and $\widehat{I}_z$ is empirical mutual information within stratum z (Wilks, 1938; Woolf, 1957). The logistic regression method denoted as LR compares nested generalized logit models, $Y \sim Z$ and $Y \sim (X, Z)$, using a likelihood-ratio test for the added predictive contribution of X (Agresti, 2013).

For a CI instance $u = (D, X, Y, Z)$, each method $m \in \mathcal{M}$ returns a p-value $p_m(u)$ and an $\alpha = 0.05$ thresholded verdict

$$v_m(u) = \mathbf{1}\{p_m(u) > 0.05\},$$

where $v_m(u) = 1$ denotes an independence verdict and $v_m(u) = 0$ denotes a dependence verdict. We focus on the *disagreement regime*

$$\mathcal{U}_{\text{dis}} = \{u : \exists m, m' \in \mathcal{M} \text{ s.t. } v_m(u) \neq v_{m'}(u)\}.$$

where at least two test methods disagree on the same CI instance. Our learning task is therefore to route within $\mathcal{U}_{\text{dis}}$: given instance features and the outputs of $\mathcal{M}$, predict the ground-truth CI label, equivalently identify which evidence source should be trusted for that instance.

3.2 Feature Space for Quantitative Envelopes

A quantitative applicability envelope should be expressed in coordinates that are statistically meaningful, computable from observed data, and interpretable after learning. Using the raw contingency tables directly would obscure this goal: their dimension changes with $|Z|$ and variable cardinality, and their entries are sensitive to arbitrary category orderings. Meanwhile, the practical difficulty of discrete CI-testing is largely driven by how finite samples are distributed across conditioning strata (Cochran, 1954; Agresti, 2013; Panzeri & Treves, 1996). A stratum groups observations with the same value of the conditioning vector Z. We define four features describing how observations populate these strata and their contingency tables. The first is conditioning **order**,

$$q_{\text{order}}(u) = |Z|,$$

which measures the fragmentation burden of stratified CI-testing. The second is **evenness**. Let $\mathcal{S}$ be the set of observed strata, n_s the number of samples in stratum s, and $p_s = n_s/N$. We define

$$q_{\text{even}}(u) = \frac{-\sum_{s\in\mathcal{S}} p_s \log p_s}{\log |\mathcal{S}|}.$$

This normalized entropy measures how evenly observations are allocated across strata (Shannon, 1948; Pielou, 1966). The third feature is **sparsity**. For each stratum s, let r_s and c_s be the locally observed cardinalities of X and Y, and let a_s be the number of observed cells in the local $r_s \times c_s$ table:

$$\text{zero}(s) = 1 - \frac{a_s}{r_s c_s}, \qquad q_{\text{sp}}(u) = \sum_{s\in\mathcal{S}} \frac{n_s}{N} \text{zero}(s).$$

This sample-weighted zero-cell rate captures the sparse-table behavior underlying expected-count rules and mutual-information bias (Cochran, 1954; Panzeri & Treves, 1996). The fourth feature is **degeneracy**. Let r and c be the global observed cardinalities of X and Y. We define

$$\deg(s) = \mathbf{1}\{r_s < r \text{ or } c_s < c\}, \qquad q_{\deg}(u) = \sum_{s\in\mathcal{S}} \frac{n_s}{N} \deg(s).$$

Degeneracy records strata in which an entire observed row or column disappears, a boundary condition that can destabilize asymptotic, exact, and regression-based inference (Agresti, 2013).

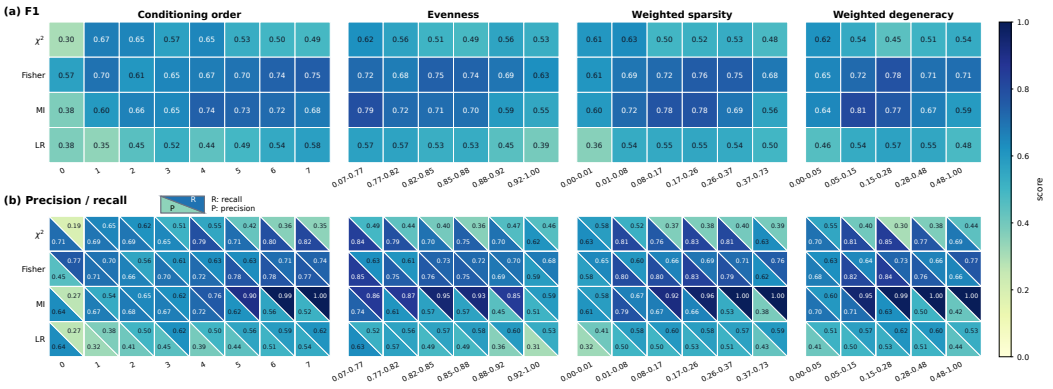


Figure 2: Feature-dependent performance of four CI-test families on disagreement instances. F1, precision, and recall vary substantially across conditioning order, evenness, weighted sparsity, and weighted degeneracy, showing that no single test is uniformly reliable.

These four coordinates define the feature space for studying method applicability. Let $y(u) \in \{0, 1\}$ denote the reference CI label and $\hat{y}(u)$ the router's prediction, with 1 denoting independence. We distinguish the test-error indicator

$$e^{(m)}(u) = \mathbf{1}\{v_m(u) \neq y(u)\}$$

from the router-overrule indicator

$$o^{(m)}(u) = \mathbf{1}\{\hat{y}(u) \neq v_m(u)\}.$$

The former measures test correctness relative to the reference label, and the latter describes the router's policy. We extract feature thresholds from changes in overrule probability and assess whether they align with changes in test error rates.

3.3 Synthetic Data Construction

Learning applicability envelopes requires data that covers regimes where CI-test reliability varies while still providing verifiable labels. We therefore construct synthetic discrete CI instances. Each instance consists of observational data D, a CI query (X, Y, Z), and a ground-truth CI label.

Graph and data generation. We build on the SiCL data-generation framework (Zhang et al., 2025), using its configurable structural causal model (SCM) pipeline and graph priors as the backbone. We extend it to discrete variables with inverse cumulative distribution function (CDF) sampling and symmetric-Dirichlet priors over conditional probability tables. Graphs are sampled from an Erdős–Rényi family with varying edge density, and node mechanisms are categorical conditional distributions sampled from symmetric Dirichlet priors.

Queries and labels. For each sampled directed acyclic graph (DAG), we generate candidate CI queries, label them by D-separation in the underlying graph (Geiger et al., 1990), evaluate all methods $m \in \mathcal{M}$, and retain the inconsistency subset for router training and envelope analysis. The D-separation labels are used as ground truth under the standard Markov and faithfulness assumptions for DAG models (Spirtes et al., 2000; Pearl, 2009; Meek, 2013). These assumptions are needed only for synthetic construction: at inference time, the router observes neither the graph nor the structural mechanisms, but only observed-data summaries and CI-test outputs. Full generation parameters are reported in Appendix A.

3.4 Preliminary Study

We first study how classical CI-test methods differ in applicability across the feature space.

Data preparation. Following the synthetic data construction pipeline in Section 3.3, we first generate 50,000 synthetic DAGs over Boolean variables and sample one CI query with observation size $n = 500$ from each DAG. We then run four representative CI-test methods on these 50,000 instances and retain the 13,641 on which disagreement emerges. To remove label imbalance as a confounder, we balance the ground truth labels to a 1:1 independence–dependence ratio, yielding 2,622 diagnostic instances. Details of the generation parameters and conditioning-order distribution are provided in Appendix A. On these 2,622 queries, Figure 2 reports each method's precision, recall, and F1 score across the proposed feature space, treating independence as the positive class.

Feature-dependent performance. Figure 2(a) shows that no method uniformly dominates the disagreement regime. Fisher is comparatively stable at higher conditioning orders, while χ^2 deteriorates as strata become fragmented. MI performs best in selected intermediate regions, and LR is consistently weaker. Variation across evenness, sparsity, and degeneracy further motivates conditioning routing decisions on these features.

Precision–recall asymmetry. Figure 2(b) shows that applicability also depends on the verdict. As order, sparsity, and degeneracy increase, MI often retains high recall for independence but loses precision, producing more false independence verdicts. In several sparse or high-order regimes, χ^2 instead loses recall and misses true independencies. Fisher is comparatively balanced. We therefore analyze overrule and error patterns separately by test-verdict direction.

4 Method: Post-training a CI-test Router

We post-train Qwen2.5-7B-Instruct (Yang et al., 2024) as a CI-test router using reinforcement learning with verifiable rewards (RLVR) (Guo et al., 2025) and supervised fine-tuning (SFT) (Ouyang et al., 2022). RLVR is our main recipe because CI labels are mechanically verifiable, allowing direct optimization of final decision correctness without relying on teacher-generated rationales; SFT serves as supervised baseline. Training details are provided in Appendix B.

4.1 Task Definition

Input. As illustrated in Figure 1(b), each prompt contains a CI query (X, Y, Z), the p-values and verdicts of four CI-test methods, and the instance-level features from Section 3.2. The router observes neither the underlying DAG nor the ground-truth d-separation label.

Output. For both RLVR and SFT, the model returns a structured response consisting of a natural-language rationale followed by a final CI verdict, `independent` or `dependent`. In RLVR, only the final verdict is rewarded. In SFT, the full teacher-generated response, including the rationale, is used as the supervised target. In evaluation, we parse the final verdict as the model prediction.

4.2 Training Data

We prompt Claude-Sonnet-4.5 for rationales and verdicts on synthetic disagreement instances and retain the 1,686 responses whose verdicts match the graph labels. RLVR and SFT share the resulting 1,348 training and 338 validation instances. RLVR uses prompts and labels and SFT also uses the teacher-generated rationales.

4.3 Training Objective

RLVR. We train with group relative policy optimization (GRPO) (Shao et al., 2024). For each prompt, multiple completions are sampled and scored by the parsed final verdict: correct decisions receive reward $+2$, while incorrect or unparsable outputs receive reward -1.

SFT. We minimize the negative log-likelihood of the teacher-generated response, including its rationale and final verdict. Filtering verifies verdict not the correctness of the rationale.

Table 1: Instance-level evaluation on balanced disagreement sets across five datasets. We report F1 score and calculate mean across datasets. Bold indicates the highest score in each column.

Method	Synthetic	child	alarm	insurance	Sachs	Mean
χ^2	0.541	0.537	0.560	0.615	0.659	0.582
Fisher/FFH	0.689	0.676	0.675	0.669	0.680	0.678
MI	0.676	0.678	0.680	0.669	0.680	0.677
LR	0.520	0.417	0.528	0.404	0.668	0.507
Majority vote	0.705	0.679	0.722	0.695	0.682	0.697
GBDT	0.720	0.701	0.744	**0.723**	**0.695**	0.717
Random forest	0.730	0.706	0.740	0.715	0.688	0.716
MLP	0.727	0.694	0.725	0.711	0.660	0.703
Logistic stacking	0.718	0.552	0.706	0.547	0.654	0.635
Decision tree	0.700	0.674	0.677	0.669	0.674	0.679
RLVR	**0.746**	**0.721**	**0.746**	**0.723**	0.691	**0.725**

5 EXPERIMENTS

We evaluate whether learned routing policy improves CI decisions, whether its boundaries identify changes in test reliability, and how these decisions affect downstream causal discovery.

5.1 EXPERIMENTAL SETUP

Evaluation data. We test transfer across five datasets without retraining. Each evaluation set retains test disagreements and balances independence and dependence labels. Synthetic data (Section 3.3) and Sachs measurements each yield 320 queries, labelled by their generating and reference graphs, respectively. Sampling 1,000 observations from each expert-built Bayesian network yields 320 queries for child (Spiegelhalter, 1992), 320 for alarm (Beinlich et al., 1989), and 216 for insurance (Binder et al., 1997). Network sizes and cardinalities are given in Appendix C.

Baselines. We compare against the four CI tests at $\alpha = 0.05$ and majority vote, which breaks 2:2 ties uniformly at random. Learned baselines include logistic stacking, a depth-3 decision tree, random forest, a multilayer perceptron (MLP) with hidden widths $(32, 16)$, and gradient-boosted decision trees (GBDT). They share the 1,348 training instances and twelve numerical inputs: four features, four p-values, and four verdicts. For these tabular routers, we use five-fold cross-validation over five seeds. SFT is evaluated on synthetic data and Sachs.

Metrics. We report F1 with independence as the positive class: precision measures the correctness of predicted independencies, and recall measures recovery of true independencies.

Sachs causal discovery setup. We replace the CI oracle inside the PC algorithm with baselines and our router, and evaluate skeleton recovery on the Sachs flow-cytometry dataset (Sachs et al., 2005). We discretize observational measurements into three levels (Ellis & Wong, 2008) and use the Fisher–Freeman–Halton (FFH) extension of Fisher's test for multi-way tables (Freeman & Halton, 1951). Against the canonical reference skeleton, we report skeleton Hamming distance (SHD), the number of missing or extra edges (Hamming, 1950), and edge precision, recall, and F1.

5.2 INSTANCE-LEVEL EVALUATION

As shown in Table 1, RLVR achieves highest overall performance. The gains therefore extend beyond choosing a fixed test or following the majority. RLVR also transfers from Boolean synthetic training data to networks with different variable cardinalities and to Sachs measurements without retraining. GBDT remains slightly stronger on Sachs (0.695 versus 0.691). The training-recipe comparison shows a larger difference on synthetic data: RLVR reaches 0.746 versus SFT's 0.620 with the same backbone and training instances. On Sachs, SFT reaches 0.700, versus RLVR's 0.691. Detailed results and uncertainty summaries appear in Appendices C.

Table 2: Recovery of CI-test applicability envelopes across five datasets, evaluated by agreement with changes in test reliability, and localization error. Bold marks the best values.

Router	Boundary recovery					Localization		
	Boundaries	Precision ↑	Recall ↑	Direction agreement ↑	False alarms ↓	Gap ↓	Null	Gap/null ↓
Decision tree	28	0.464	0.325	13/19 (68%)	9 (45%)	**0.054**	0.330	**0.17**
RLVR (ours)	17	**0.588**	0.250	**10/12 (83%)**	**5 (25%)**	0.096	0.236	0.41
MLP	22	0.409	0.225	9/17 (53%)	**5 (25%)**	0.107	0.199	0.54
GBDT	16	0.375	0.150	6/11 (55%)	**5 (25%)**	0.155	0.167	0.93
Random forest	17	0.412	0.175	7/11 (64%)	6 (30%)	0.211	0.188	1.12
Logistic stacking	36	0.500	**0.450**	18/29 (62%)	7 (35%)	0.260	0.213	1.22

5.3 Applicability Envelope Analysis

An applicability envelope describes the conditions under which a test verdict can be trusted. We characterize envelopes feature by feature: thresholds summarize candidate boundaries, and labelled test errors assess whether they reflect reliability changes. For test m and feature f, we fit the same Bernoulli change-point model to overrule indicators $o_i^{(m)}$ and error indicators $e_i^{(m)}$ (Section 3.2):

$$\Pr(z_i = 1 \mid f_i) = p_0 \mathbf{1}\{f_i < \tau\} + p_1 \mathbf{1}\{f_i \geq \tau\}, \qquad z_i \in \{o_i^{(m)}, e_i^{(m)}\}.$$

The threshold τ separates probabilities p_0, p_1. Maximum-likelihood fits give router and reference boundaries $\tau_r, \tau_{\text{ref}}$, retained when the Bayesian information criterion (BIC) favors two rates over one after penalizing complexity. Reference fits use unfiltered labelled pools without router predictions (Appendix E). We compare errors and overrides of the same verdict type, such as independence. *Direction agreement* requires errors and overrules to increase toward the same side.

We evaluate router boundaries across 60 dataset–test–feature combinations, 40 with reference boundaries. Precision and recall measure directionally correct matches relative to all router and reference boundaries. *Direction agreement* considers only combinations with both boundaries. *False-alarm rate* measures boundary detection in the 20 combinations without reference boundaries. For directionally correct matches, *gap* is the mean distance normalized by the feature range, *null* is its random-threshold counterpart on the same combinations. *Gap/null* below 1 indicates better localization than random placement. Appendix E gives calculation details.

Validating learned applicability. Table 2 reveals different strengths among routers. RLVR achieves the highest direction agreement and precision, with a false-alarm rate tied for the lowest. The decision tree localizes its directionally correct boundaries more accurately, while logistic stacking recovers more reference transitions. For interpreting an applicability envelope, direction is more fundamental than exact localization. A correctly directed but imperfectly located boundary can still provide useful guidance away from the transition. An incorrectly directed boundary instead recommends trusting the test on its less reliable side, even if its threshold is precisely located. RLVR's combination of strong direction agreement, high precision, and localization better than random placement therefore makes its recovered envelopes useful guides to test applicability, although they cover only part of the reference transitions. More broadly, this analysis provides a verifiable way to characterize the envelopes learned by different routers, distinguishing whether they identify the correct reliability trend, where the transition occurs, and how much of the reference envelope structure they recover.

Envelopes change across regimes. We keep the router fixed and repeat the envelope analysis at different sample sizes. The reference χ^2 order threshold is 3 at $N = 100$, 5 at both $N = 200$ and 350, and 6 at $N = 500$, showing that the transition moves to higher conditioning orders as more observations become available. Across these settings, the router's mean normalized gap is 0.143, compared with 0.222 for random thresholds. On Sachs, where each variable is discretized into three categories, the router matches the χ^2 order boundary at 5; its mean gap over order, sparsity, and evenness is 0.063 versus a random baseline of 0.100. These results show that the numerical boundaries depend on the data regime, while the fixed router continues to identify changes in test reliability without retraining (Appendix F).

Table 3: PC skeleton recovery on Sachs. Bold marks the lowest SHD and highest precision and F1; several methods tie on recall.

Method	SHD ↓	Precision	Recall	F1
χ^2	14	0.6154	0.4706	0.5333
Fisher/FFH	14	0.6364	0.4118	0.5000
LR	20	0.4118	0.4118	0.4118
MI	13	0.7000	0.4118	0.5185
Majority vote	13	0.6667	0.4706	0.5517
GBDT	12	0.7273	0.4706	0.5714
SFT	12	0.7273	0.4706	0.5714
RLVR	**11**	**0.8000**	0.4706	**0.5926**

What an envelope revealed. As an example, the learned MI envelope points to a specific behavior: as conditioning strata lose category support, the router increasingly overrules MI's independence verdicts. The extracted boundaries occur at sparsity 0.011 and degeneracy 0.580. Investigating this behavior revealed a mismatch in the tested MI implementation, which sums per-stratum G^2 statistics and degrees of freedom. If either variable has only one observed category in a stratum, every nonzero term in the statistic reduces to zero. That stratum therefore contributes nothing to G^2. Yet when its table retains the global r row and c column categories, it can still contribute $(r-1)(c-1)$ degrees of freedom. As collapsed strata accumulate, they add degrees of freedom without adding evidence of dependence. The pooled p-value increases, making MI more likely to report independence.

This mechanism explains why the router becomes less willing to trust MI's independence verdicts as degeneracy increases. In the diagnostic pool, an average of 12 of 22 strata per query are collapsed. Recomputing degrees of freedom from each stratum's observed category support changes 16.5% of verdicts, all from independence to dependence, and raises accuracy from 0.601 to 0.707. The changes are consistent with the identified mechanism. The envelope thus provided more than a description of routing behavior: it identified where to inspect the test and led to an implementation issue whose effect could be checked independently of the router. This diagnosis applies to implementations that retain nominal degrees of freedom for collapsed strata (Appendix G).

5.4 Real-world Causal Discovery

Table 3 shows that the router's instance-level gains also translate to downstream causal discovery. Replacing the CI oracle in PC with the RLVR router yields the lowest skeleton Hamming distance and the highest skeleton F1. This result is important because CI-test instances are generated adaptively by PC's skeleton-search procedure rather than sampled from the held-out CI-instance set. A small number of CI decisions can therefore change the recovered graph. The improvement suggests that the learned applicability envelopes remain useful when CI-test routing is embedded in a downstream pipeline. Appendix I visualizes the recovered skeletons and shows that RLVR recovers a biologically meaningful membrane-signaling structure. The relevant edge decisions arise from method-disagreement regimes, supporting the practical relevance of our formulation. During evaluation router answers 25 of 94 PC queries (26.6%), the remaining 69 use test consensus.

6 Limitations

This study focuses on discrete CI testing in disagreement regimes. The recovered envelopes characterize a subset of reliability transitions under the evaluated data conditions. Extending this analysis to continuous and mixed data would broaden its scope. Downstream validation currently uses Sachs with graph-derived CI labels, motivating evaluation on additional networks and reference structures. A further step is to assess the complete testing pipeline, including errors shared by unanimous tests, and establish whether nominal Type-I error rates are maintained. Appendix F discusses additional sources of variation.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing and debugging parts of the training and evaluation code. We did not use generative AI tools to develop theoretical models, formulate or prove mathematical claims, propose hypotheses, design experiments, clean or reformat data, analyze data, or interpret results. Translation, and qualitative or thematic analysis do not apply to this work. Additionally, we used generative AI tools for formatting, and help search for related work. We have reviewed all AI-assisted work. We checked every suggested reference against the original source, and we reviewed every AI-suggested wording change to make sure it kept our intended meaning. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies the reliability of conditional independence testing using synthetic data, established Bayesian-network benchmarks, and previously published Sachs flow-cytometry measurements. It does not involve recruiting participants or collecting new human-subject data. We cite the sources of the datasets, models, and software used in the study; relevant licensing information is provided in Appendix J. The proposed router supports statistical analysis, and its outputs should be interpreted within the assumptions and data conditions of the evaluation. In particular, improved CI prediction or graph recovery does not establish the validity of individual causal claims. The Sachs experiment evaluates recovery against a reference graph and is not intended to support clinical decisions. The paper reports the evaluation scope, shared-test failures, and limitations of the recovered applicability envelopes to support appropriate interpretation and use.

REPRODUCIBILITY STATEMENT

We provide the experimental procedures and settings in the main paper and appendices. Appendix A describes synthetic data generation, graph-derived CI labels, query filtering, and label balancing. Appendix B documents the prompt format, supervised-response selection, training and inference settings, and computing resources. Appendix C provides additional evaluation details, network-transfer accuracy, and uncertainty summaries. Appendix D describes the Sachs evaluation, while Appendices E, F, and G explain envelope estimation and validation, evaluation across data regimes, and the MI diagnostic. Recovered skeletons and selected downstream CI decisions appear in Appendix I. An anonymized code repository is linked in the abstract.

REFERENCES

Alan Agresti. *Categorical data analysis*. John Wiley & Sons, 2013.

Ingo A Beinlich, Henri Jacques Suermondt, R Martin Chavez, and Gregory F Cooper. The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks. In *AIME 89: Second European Conference on Artificial Intelligence in Medicine, London, August 29th–31st 1989. Proceedings*, pp. 247–256. Springer, 1989.

Richard Bellman. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6):503–515, 1954.

John Binder, Daphne Koller, Stuart Russell, and Keiji Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29(2):213–244, 1997.

Ronald D Boschloo. Raised conditional level of significance for the $2\times$ 2-table when testing the equality of two probabilities. *Statistica neerlandica*, 24(1):1–9, 1970.

Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luján. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *The journal of machine learning research*, 13:27–66, 2012.

William G Cochran. Some methods for strengthening the common χ 2 tests. *Biometrics*, 10(4): 417–451, 1954.

Haoyue Dai, Rui Ding, Yuanyuan Jiang, Shi Han, and Dongmei Zhang. Ml4c: Seeing causality through latent vicinity. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pp. 226–234. SIAM, 2023.

A Philip Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 41(1):1–15, 1979.

Byron Ellis and Wing Hung Wong. Learning causal bayesian network structures from experimental data. *Journal of the American Statistical Association*, 103(482):778–789, 2008. ISSN 01621459. URL http://www.jstor.org/stable/27640100.

Ronald A Fisher. On the interpretation of χ 2 from contingency tables, and the calculation of p. *Journal of the royal statistical society*, 85(1):87–94, 1922.

GH Freeman and John H Halton. Note on an exact treatment of contingency, goodness of fit and other problems of significance. *Biometrika*, 38(1/2):141–149, 1951.

Dan Geiger, Thomas Verma, and Judea Pearl. d-separation: From theorems to algorithms. In *Machine intelligence and pattern recognition*, volume 10, pp. 139–148. Elsevier, 1990.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Richard W Hamming. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160, 1950.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Christopher Meek. Strong completeness and faithfulness in bayesian networks. *arXiv preprint arXiv:1302.4973*, 2013.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Stefano Panzeri and Alessandro Treves. Analytical estimates of limited sampling biases in different information measures. *Network: Computation in neural systems*, 7(1):87, 1996.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. Gorilla: Large language model connected with massive apis. *Advances in Neural Information Processing Systems*, 37: 126544–126565, 2024.

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

Karl Pearson. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, 1900.

Evelyn C Pielou. The measurement of diversity in different types of biological collections. *Journal of theoretical biology*, 13:131–144, 1966.

John R Rice. The algorithm selection problem. In *Advances in computers*, volume 15, pp. 65–118. Elsevier, 1976.

Jakob Runge. Conditional independence testing based on a nearest-neighbor estimator of conditional mutual information. In *International Conference on Artificial Intelligence and Statistics*, pp. 938–947. Pmlr, 2018.

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

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

Rajat Sen, Ananda Theertha Suresh, Karthikeyan Shanmugam, Alexandros G Dimakis, and Sanjay Shakkottai. Model-powered conditional independence test. *Advances in neural information processing systems*, 30, 2017.

Rajen D Shah and Jonas Peters. The hardness of conditional independence testing and the generalised covariance measure. 2020.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Kate A Smith-Miles. Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys (CSUR)*, 41(1):1–25, 2009.

David J Spiegelhalter. Learning in probabilistic expert systems. *Bayesian statistics*, 4:447–465, 1992.

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

Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*, 2023.

Samuel S Wilks. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The annals of mathematical statistics*, 9(1):60–62, 1938.

Barnet Woolf. The log likelihood ratio test (the g-test). *Annals of human genetics*, 21(4):397–409, 1957.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024. URL `https://api.semanticscholar.org/CorpusID:274859421`.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Jiaru Zhang, Rui Ding, Qiang Fu, Bojun Huang, Zizhen Deng, Yang Hua, Haibing Guan, Shi Han, and Dongmei Zhang. Learning identifiable structures helps avoid bias in dnn-based supervised causal learning. *arXiv preprint arXiv:2502.10883*, 2025.

Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Kernel-based conditional independence test and application in causal discovery. *arXiv preprint arXiv:1202.3775*, 2012.

A SYNTHETIC CI-INSTANCE CONSTRUCTION

Generation parameters. Table 4 summarizes the synthetic data-generation settings used in the main experiments. We build on the SiCL data-generation framework (Zhang et al., 2025), extending it to discrete variables with inverse-CDF categorical mechanisms and symmetric-Dirichlet conditional probability tables. The main training and evaluation splits use Boolean variables so that all four CI-test families are directly comparable.

Table 4: Synthetic CI-instance generation settings.

Component	Setting
Graph family	Erdős–Rényi DAGs with varying edge density
Mechanism	Categorical SCM with inverse-CDF sampling
CPT prior	Symmetric Dirichlet, $\alpha = 1.0$
Main variable cardinality	Boolean
Observation type	Observational, no interventions
Ground-truth label	D-separation in the sampled DAG
Main sample size	$n = 500$
Filtering regime	Four-method disagreement cases

Diagnostic disagreement set. For the preliminary study, we generate $50{,}000$ synthetic DAGs and sample one Boolean CI query from each DAG at $n = 500$. Running the four statistical CI tests yields $13{,}641$ instances with method disagreement. We then balance the d-separation labels to a $1{:}1$ independence–dependence ratio, giving $2{,}622$ diagnostic instances. Their conditioning-order distribution is shown in Table 5.

Table 5: Conditioning-order distribution in the balanced $2{,}622$-instance diagnostic disagreement set.

Order	0	1	2	3	4	5	6	7
Count	52	138	248	234	272	426	518	734

B MODEL TRAINING AND PROMPT DETAILS

Prompt format. Both SFT and RLVR use the same JSON-style prompt. The raw contingency tables are omitted in the reported no-table setting. A representative prompt is:


```
{
 "task": "Integrate conflicting CI-test outputs.",
 "variables": {
   "x": "X4",
   "y": "X7",
   "z": ["X1", "X8", "X9", "X5", "X0", "X2", "X6"]
 },
 "inconsistency": {
   "votes": {"independent": 1, "dependent": 3},
   "by_method": {
     "chi2": false,
     "logistic_regression": false,
     "fisher": false,
     "mutual": true
   },
   "num_ok": 4
 },
 "features": {
   "order": 7,
   "evenness": 0.8288,
   "sparsity": 0.0325,
```

```
    "degenerate": 0.836
  },
  "methods": {
    "chi2": {
      "ok": true,
      "is_independent": false,
      "p_value": 0.013836,
    },
    "logistic_regression": {
      "ok": true,
      "is_independent": false,
      "p_value": 0.0000755,
    },
    "fisher": {
      "ok": true,
      "is_independent": false,
      "p_value": 0.019802,
    },
    "mutual": {
      "ok": true,
      "is_independent": true,
      "p_value": 0.999954,
    }
  },
  "output_format": {
    "rationale": "On which evidence and why, think step-by-step",
    "decision": "one of {\"independent\",\"dependent\"}"
  }
}
```


The model output is parsed as JSON whenever possible; evaluation uses only the final `decision` field.

SFT data construction. We prompt Claude-Sonnet-4.5 to generate a rationale and final verdict for each training instance using the same input format. Since these responses are not mechanically guaranteed to be correct, we retain only examples whose generated final verdict matches the d-separation label. This yields $1{,}686$ supervised examples, split into $1{,}348$ training and 338 validation examples. SFT is full-parameter fine-tuning of Qwen2.5-7B-Instruct with bfloat16 precision, cosine learning-rate scheduling, learning rate 2×10^{-5}, weight decay 0.01, warmup ratio 0.05, maximum sequence length 512, two epochs, per-device batch size 8, and gradient accumulation 4.

RLVR training. The RLVR router uses the same $1{,}348$ training prompts and verifiable d-separation labels. We train Qwen2.5-7B-Instruct with GRPO (Shao et al., 2024), initialized from the base model for the main RLVR recipe. The reward is verdict-level: $+2$ for a parsed decision matching the d-separation label and -1 for an incorrect or unparsable decision. We use one epoch, learning rate 10^{-6}, per-device batch size 1, gradient accumulation 4, 8 completions per prompt, maximum prompt length 1800, maximum completion length 512, bfloat16 precision, gradient checkpointing, and DeepSpeed ZeRO-3.

Inference settings. For reported local-router evaluations, we use deterministic decoding with a maximum of 512 new tokens and parse the final `decision` field.

Computing resources. Model post-training was run on a single machine with 8 NVIDIA A100 GPUs and took approximately 3 hours for the reported training runs. All local router inference was run on 4 NVIDIA Tesla T4 GPUs. Evaluating the 320-instance CI-test split takes approximately 1.5 hours end-to-end, including prompt construction, model generation, and verdict parsing.

C Additional Instance-level Evaluation

The additional Bayesian networks differ in size and cardinality: child has 20 variables with 2–6 levels, alarm has 37 with 2–4 levels, and insurance has 27 with 2–5 levels.

Network transfer. For child, alarm, and insurance, we sample 1,000 observations from the published conditional probability tables, draw 4,000–8,000 queries stratified by conditioning order, assign d-separation labels, retain four-test disagreements, and balance labels 1:1. These datasets use expert-built mechanisms but remain sampled data rather than additional real-world measurements. Relating graph labels to population CI relies on the Markov and faithfulness assumptions described in Section 3.3.

Matched routers and mean F1. The five tabular routers use the same twelve numerical inputs and 1,348 training instances, with five-fold cross-validation over five seeds. The LLM prompt additionally includes query identifiers, method names, and formatting. Matching numerical inputs therefore does not isolate the effects of prompt semantics or pretrained knowledge. The same trained GBDT is used throughout the evaluations.

For each method m, the main table reports

$$\overline{\mathrm{F1}}_m = \frac{1}{5}\sum_{d=1}^{5} \mathrm{F1}_{m,d}.$$

Each dataset receives equal weight, including insurance, which has fewer evaluation queries. The mean is calculated from the displayed per-dataset scores; it is neither pooled F1 nor an uncertainty estimate. An always-independent predictor has F1 $2/3$ and accuracy $1/2$ on these balanced sets.

C.1 Network-transfer Accuracy and Uncertainty

Table 6: Accuracy on balanced disagreement sets from three Bayesian networks.

Method	child	alarm	insurance
RLVR	0.644	0.681	0.634
GBDT	0.635	0.677	0.639
Majority vote	0.628	0.684	0.630
χ^2	0.575	0.613	0.606
Fisher/FFH	0.625	0.675	0.625
MI	0.531	0.541	0.509
LR	0.537	0.547	0.509

Table 7: Network-transfer uncertainty for RLVR and two comparison methods. Brackets are instance-bootstrap 95% intervals for RLVR and majority vote, and ranges over five seeds for GBDT.

Network	Method	F1 [interval]	Accuracy [interval]
child	RLVR	0.721 [0.667, 0.768]	0.644 [0.591, 0.697]
	GBDT	0.701 [0.679, 0.724]	0.635 [0.622, 0.647]
	Majority vote	0.679 [0.622, 0.732]	0.628 [0.575, 0.681]
alarm	RLVR	0.746 [0.695, 0.793]	0.681 [0.628, 0.731]
	GBDT	0.744 [0.738, 0.751]	0.677 [0.669, 0.688]
	Majority vote	0.722 [0.665, 0.770]	0.684 [0.631, 0.731]
insurance	RLVR	0.723 [0.662, 0.779]	0.634 [0.569, 0.699]
	GBDT	0.723 [0.717, 0.731]	0.639 [0.620, 0.653]
	Majority vote	0.695 [0.628, 0.755]	0.630 [0.565, 0.694]

Instance-bootstrap intervals and seed ranges quantify different sources of variation. Query-level resampling does not capture all variation from resampling an entire observational dataset or selecting a different network.

D Sachs Evaluation Details

Exact test for multi-level Sachs data. The synthetic benchmark is Boolean, where Fisher's exact test is applied to 2×2 contingency tables. The Sachs flow-cytometry data are discretized into three

levels, so per-stratum tables can be larger than 2×2. We therefore use the Fisher–Freeman–Halton extension for $r \times c$ tables. For computational tractability, p-values are estimated by Monte Carlo sampling tables with fixed margins using the Patefield sampler. The two-sided p-value uses probability ordering under the fixed-margin multivariate hypergeometric distribution:

$$p = \Pr\{P(T \mid \text{margins}) \leq P(T_{\text{obs}} \mid \text{margins})\}.$$

For effective 2×2 tables, we use the standard Fisher exact test.

PC setup. For the downstream Sachs experiment, each continuous protein measurement is discretized into three levels and PC is run on the observational condition. The recovered skeleton is compared against the observational Sachs reference skeleton used in the main text. This PC experiment is separate from the 320 sampled Sachs CI-instance transfer set: PC generates CI queries adaptively during skeleton search. Every query runs the four tests. Unanimous verdicts use the consensus shortcut, and disagreements among at least two successful tests invoke the router. The reported run contains 94 queries, including 25 router calls and 69 consensus decisions.

E Analysis Details of Applicability Envelopes

Change-point estimation. For a method and feature coordinate, the change-point model in Section 5.3 is fitted separately to router-overrule and test-error indicators. Candidate thresholds are drawn from observed feature values, subject to minimum support on both sides. For each threshold, the two Bernoulli rates are estimated by the empirical indicator means below and above it.

The Bayesian information criterion (BIC) compares fit while penalizing model complexity. For maximized log-likelihood ℓ, sample size n, and parameter count k, $\text{BIC} = k \log n - 2\ell$. The constant-rate model has one rate parameter; the change-point model has two rates and a threshold. We retain a transition when

$$\Delta\text{BIC} = \text{BIC}_{\text{change point}} - \text{BIC}_{\text{constant}} < 0.$$

Comparisons pair the same test-verdict type (independence or dependence). This differs from direction agreement, which compares the signs of the error-rate and overrule-rate changes across the feature threshold. Reference and router pools differ in disagreement filtering and label balance, so the fitted conditional rates need not coincide. Reference transitions are estimated without router predictions from unfiltered labelled pools of 8,000 synthetic instances, 3,000 Sachs queries, and 4,000 queries per bnlearn network. They remain empirical model-based estimates rather than known population discontinuities.

Detection denominators. Let $\mathcal{C}_+$ and $\mathcal{C}_-$ denote the reference-supported and unsupported cells, with sizes 40 and 20. For router claims $\mathcal{A}_r$, let $\mathcal{H}_r$ contain claims in $\mathcal{C}_+$ whose overrule-increase direction matches the reference error-increase direction. Then

$$\text{precision} = \frac{|\mathcal{H}_r|}{|\mathcal{A}_r|}, \qquad \text{recall} = \frac{|\mathcal{H}_r|}{40}, \qquad \text{false alarm rate} = \frac{|\mathcal{A}_r \cap \mathcal{C}_-|}{20}.$$

For RLVR, 12 claims fall in reference-positive cells, and 10 match the reference direction. Five additional claims fall in reference-negative cells. Direction agreement is therefore $10/12$, precision is $10/17$, and recall is $10/40$.

Localization comparator. For each directional hit c,

$$g_c = \frac{|\tau_r - \tau_{\text{ref}}|}{R_f}, \qquad \nu_c = \frac{1}{|S_c|} \sum_{u \in S_c} \frac{|f(u) - \tau_{\text{ref}}|}{R_f},$$

where S_c is the comparison subpopulation. The scale R_f is the feature range: 7 for conditioning order (0–7) and 1 for evenness, sparsity, and degeneracy (0–1). A gap of 0.1 is therefore one tenth of the corresponding feature range. The null averages the gap from a threshold sampled uniformly over the instances' observed feature values. Reported gap/null values are ratios of mean gap to mean null over each router's directional hits.

Different routers have different hit sets, which limits direct comparison of conditional localization averages. BIC support is not a multiple-testing-adjusted guarantee; lack of support does not establish an absent population transition.

Table 8: Direction and localization summaries. Direction uses claims in reference-positive cells; localization uses only directionally matching hits. Claims include all router-supported cells.

Router	Claims	Direction matches	Mean gap	Mean null	Gap/null
Decision tree	28	13/19	0.054	0.330	0.17
RLVR	17	10/12	0.096	0.236	0.41
MLP	22	9/17	0.107	0.199	0.54
GBDT	16	6/11	0.155	0.167	0.93
Random forest	17	7/11	0.211	0.188	1.12
Logistic stacking	36	18/29	0.260	0.213	1.22

Original synthetic thresholds. Table 9 retains the original router-overrule thresholds. They characterize one synthetic regime and are not universal test-validity cutoffs. LR has an overall overrule rate of 0.594; Fisher has no BIC-supported single-feature threshold in this analysis.

Table 9: BIC-supported router-overrule thresholds in the original synthetic regime. Negative ΔBIC favors a change point.

Method	Feature	Threshold	ΔBIC
χ^2	Order	5	-20.98
χ^2	Sparsity	0.037	-22.09
χ^2	Evenness	0.921	-3.20
χ^2	Degeneracy	0.039	-38.72
LR	Order	5	-16.05
LR	Sparsity	0.011	-14.18
LR	Degeneracy	0.033	-25.13
MI	Sparsity	0.011	-3.21
MI	Degeneracy	0.580	-1.54

Relation to expected-count guidance. The order threshold is qualitatively consistent with reduced contingency-table support as conditioning order increases. However, average observations per stratum do not determine the distribution of null-expected cell counts. The extracted thresholds describe empirical router behavior rather than an exact recovery or relaxation of an expected-count criterion (Cochran, 1954).

Scope of the comparisons. Cross-network comparisons do not independently control graph density, and Dirichlet concentration was not systematically varied. Further analysis is needed to separate router error, sampling, and the extraction criterion as causes of incomplete boundary recovery.

F Threshold Variation Across Regimes

The router remains fixed while reference and router boundaries are re-estimated at each sample size. This analysis summarizes router-supported comparisons among reference transitions; it does not enumerate reference-negative cells or estimate a false-alarm rate.

Table 10: Sample-size sensitivity. Supported comparisons include direction mismatches and differ from the directional hits defining recall in Table 2.

Setting	Reference transitions	Supported comparisons	Mean gap	Direction matches
$N = 100$	13	9	0.222	6/9
$N = 200$	14	13	0.138	9/13
$N = 350$	15	10	0.111	10/10
$N = 500$	15	14	0.119	11/14
Pooled	57	46	0.143	36/46

The pooled null gap is 0.222. Reference χ^2 order thresholds are 3, 5, 5, and 6 across increasing sample sizes. The reported LR degeneracy boundary varies from 0.785 to 0.005 across these settings.

For three-level Sachs, 14 reference transitions and 11 router-supported comparisons are reported. The mean gap over order, sparsity, and evenness is 0.063, compared with a null of 0.100. The χ^2 order threshold is matched at 5; its sparsity and evenness thresholds are within 0.021 of their references. Degeneracy is considered separately because category loss relative to global support is not directly scale-comparable across cardinalities.

The additional networks extend evaluation to 2–6-level variables and different conditional probability tables. Graph structure, density, cardinality, and mechanisms change together, so these comparisons do not isolate density sensitivity. No controlled Dirichlet-concentration sweep was performed.

G MI Degrees-of-Freedom Diagnostic

The router thresholds at sparsity 0.011 and degeneracy 0.580 motivate this diagnostic. For stratum s, the tested statistic is

$$G_s^2 = 2 \sum_{x,y:n_{sxy}>0} n_{sxy} \log \frac{n_{sxy} n_s}{n_{sx+} n_{s+y}}, \qquad G^2 = \sum_s G_s^2.$$

Here n_{sxy} is the cell count, n_{sx+} and n_{s+y} are row and column totals, and n_s is the stratum size. If either observed margin collapses to one category, every surviving ratio is one and $G_s^2 = 0$. Under a global-category table construction, the stratum may nevertheless contribute $(r-1)(c-1)$ nominal degrees of freedom, where r and c are the global observed category counts of X and Y. Holding the statistic fixed while increasing its reference degrees of freedom increases the upper-tail p-value.

In the reported diagnostic pool, an average of 12 of 22 strata per query are collapsed. Recomputing per-stratum effective degrees of freedom changes 16.5% of decisions, all from independence to dependence, and raises reported accuracy from 0.601 to 0.707. The count of collapsed strata is related to, but distinct from, the sample-weighted degeneracy feature. For multilevel variables, losing a category need not collapse an entire margin.

This diagnostic concerns the implementation evaluated here. Sample zeros need not be structural zeros, and improved classification accuracy does not establish Type-I error control. No corrected-MI router or downstream PC comparison is reported.

H Natural Query Distribution and Shared Failures

The unfiltered analysis uses 8,000 synthetic queries without imposing label balance. The reported frequency of true independence is 0.233. Test disagreements occur in 28.2% of queries; unanimous verdicts occur in 71.8%.

Table 11: Shared failures on the unfiltered synthetic query pool. Error fractions condition on the indicated unanimous verdict.

Unanimous verdict	Queries	Incorrect	Incorrect fraction
Independent	3,590	1,970	54.9%
Dependent	2,155	12	0.6%
Total	5,745	1,982	34.5%

The mean feature profiles, ordered as conditioning order, evenness, sparsity, and degeneracy, are $(2.76, 0.876, 0.089, 0.110)$ for jointly incorrect verdicts and $(3.25, 0.858, 0.096, 0.097)$ for jointly correct verdicts. Similarity of these means does not establish that the groups cannot be separated. Resolving disagreements alone does not address the shared errors preserved by the consensus gate.

I Sachs Skeleton Case Study

Skeleton interpretation. The Sachs skeleton visualization shows that the RLVR router preserves the plcg–PIP2–PIP3 structure, which is part of the phosphoinositide membrane-signaling module.

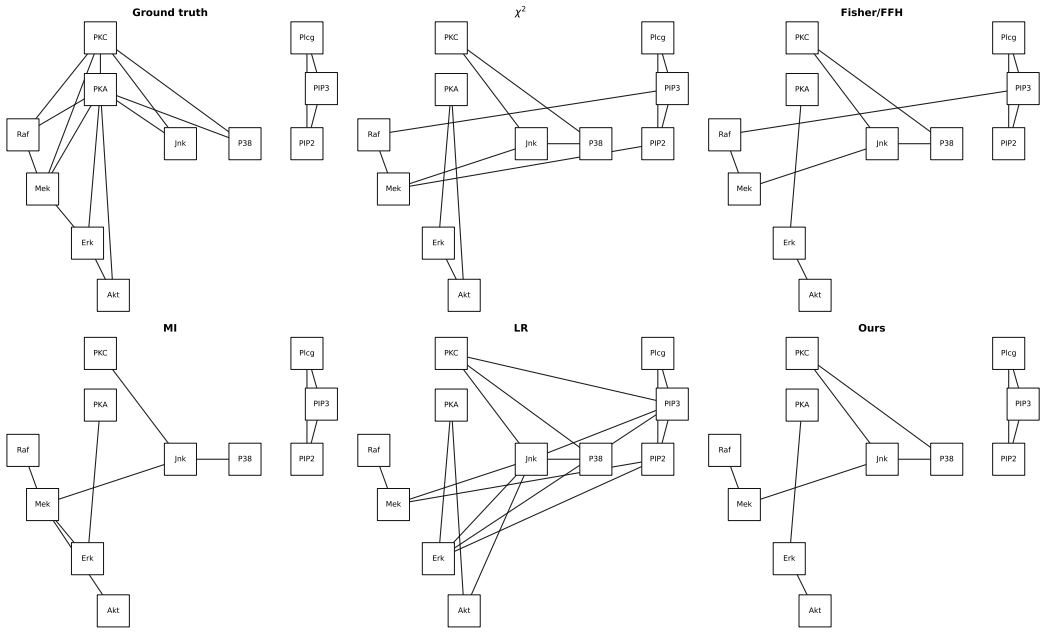


Figure 3: Recovered Sachs skeletons from PC with different CI oracles, together with the observational Sachs reference skeleton. RLVR achieves the lowest SHD and highest skeleton F1 among the evaluated variants and preserves the local plcg–PIP2–PIP3 membrane-signaling structure.

This local structure is recovered by MI and the router, but the router obtains a better global skeleton score by avoiding additional errors elsewhere. This visualization illustrates downstream routing behavior; it does not independently validate the extracted feature thresholds.

Edge-removal disagreements. Table 12 shows two PC edge-removal decisions around this module. In both cases, the statistical methods split 2:2 and the instance has elevated weighted degeneracy. The router returns independence and removes the edge in both cases. These examples illustrate disagreement resolution during adaptive search without isolating the causal contribution of degeneracy.

Table 12: Two Sachs PC edge-removal decisions where the router resolves 2:2 statistical-test disagreement. Values in parentheses are p-values.

Pair	Separating set	$q_{\deg}$	χ^2	LR	Fisher/FFH	MI / Ours
$\mathrm{pmek}, \mathrm{PIP2}$	$\{\mathrm{praf}, \mathrm{PKA}\}$	0.652	dep (0.011)	dep (0.014)	indep (0.095)	indep (0.763) / indep
$\mathrm{praf}, \mathrm{PIP3}$	$\{\mathrm{pmek}\}$	0.493	dep (0.050)	indep (0.176)	dep (0.048)	indep (0.215) / indep

J Assets and Licenses

We use the Sachs flow-cytometry data and reference graph from the bnlearn Bayesian Network Repository, distributed under CC BY-SA 3.0, and cite the original Sachs et al. study (Sachs et al., 2005). Our synthetic data pipeline builds on SiCL (Zhang et al., 2025), released under the MIT License. We post-train Qwen2.5-7B-Instruct (Yang et al., 2024), whose model card reports an Apache-2.0 license.