Adaptive Hypothesis Testing for Dependent Graph-Valued Structures via Topological Value of Information

Anonymous Authors[1]

Abstract

Modern ML systems increasingly output uncertain graph-valued structures, including causal graphs, mechanistic graphs, interaction networks, and graph-based explanations. In these settings, reliability depends on deciding which dependent structural hypotheses to test next under a limited budget. We formulate this as *adaptive structural hypothesis testing*, where candidate edges are binary hypotheses, tests produce noisy evidence, and test value depends on the global graph-valued hypothesis state. We introduce *Topological Complexity Minimization* (TCM), a cost-aware adaptive testing policy that scores a probabilistic belief graph using connected components (β_0) and independent cycles (β_1), and selects tests that maximize expected topological simplification per unit cost. We evaluate TCM on standard Bayesian network DAGs from `bnlearn` (CHILD, INSURANCE) and 200 atom-level mechanistic graphs from `mech-USPTO-31k`. Across cycle-dominated, fragmentation-dominated, and mixed corruption regimes, TCM achieves the highest structural efficiency in both benchmark families while preserving competitive edge-level F1. On mechanistic graphs, TCM improves over CYCLERANK from 0.452 to 0.576 in fragmentation regimes and from 0.715 to 0.830 in mixed regimes; on the Bayesian network DAGs, the corresponding gains are 1.062 to 1.471 and 1.406 to 2.049. These results show that local edge uncertainty is not always aligned with graph-level disambiguation, and that lightweight topological utilities can support adaptive hypothesis testing under structural dependence.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

1. Introduction

Modern machine learning systems increasingly produce uncertain *relational structures*: causal graphs, mechanistic graphs, interaction graphs, and graph-valued explanations. In these settings, the central question is not only how uncertain a prediction is, but *which structural hypotheses should be tested next*. Many scientific models can be represented as graphs whose nodes are entities and whose edges are candidate interactions, causal relations, or mechanistic steps (1; 2; 3; 4; 5). Scientific progress then depends on adaptively selecting measurements, computations, or queries that most efficiently reduce uncertainty over the underlying graph.

We study this problem as *adaptive structural hypothesis testing*. For each candidate edge (i, j), the null and alternative hypotheses are

$$H_{ij}^0 : e_{ij} = 0 \qquad \text{versus} \qquad H_{ij}^1 : e_{ij} = 1,$$

where e_{ij} denotes whether the relation exists in the unknown ground-truth structure. Unlike classical multiple-testing settings where hypotheses are often treated as exchangeable or weakly dependent, edge hypotheses in a graph are structurally coupled: resolving one edge can merge disconnected components, remove spurious cycles, or change the interpretation of other candidate relations. The testing problem is therefore not only local—which edge is most uncertain?—but structural: which test most reduces global ambiguity?

Most uncertainty methods in scientific ML quantify aleatoric uncertainty in observations or epistemic uncertainty over parameters within a fixed relational structure (6; 7; 8; 9; 10; 11; 12; 13; 14; 15). Here we focus instead on **structural epistemic uncertainty**: uncertainty about *which edges exist and how they are organized*. This uncertainty is reducible by information acquisition and is closely related to uncertainty over causal or mechanistic structure (1; 2; 16; 17; 18; 19).

Graph-based world models and graph neural networks are standard tools for representing molecules, materials, and dynamical systems (20; 3; 21; 4; 5). Related work studies Bayesian network structure learning, causal discovery, and partially observed interaction graphs (22; 23; 24; 25; 26; 27; 28). These methods typically learn graph structure end-to-end or evaluate recovered structures after the fact; they do

not define an explicit decision-layer rule for selecting which structural hypothesis to test next under a limited budget.

Our motivating application is mechanistic reasoning for organic reactions. Existing ML approaches predict products, retrosynthetic disconnections, or mechanistic electron-flow steps (29; 30; 31; 32; 33; 34; 35; 36; 37; 38; 39; 40; 41; 42; 43; 44; 45). They optimize predictive accuracy or sequence recovery, but do not choose which uncertain mechanistic relation should be queried next under a cost constraint.

Our problem is related to Bayesian experimental design, active learning, and adaptive experimentation (46; 47; 48; 49; 50; 51; 52; 53; 54; 55; 56; 57). In chemistry and materials science, these ideas underpin adaptive design and closed-loop discovery (21; 58; 59; 60). However, most objectives are defined over scalar properties or predictive uncertainty for a fixed model. Here, the hypotheses themselves are structural: the unknowns are which edges exist and how they connect.

We propose to use simple topological summaries as decision-layer utilities. Topological descriptors such as Betti numbers and persistent homology summarize complex graphs and point clouds (61; 62; 63; 64), with applications in chemistry, materials science, and machine learning (65; 66; 67; 68; 69; 70; 71; 72). Unlike prior uses of topology as a feature, prior, or regularizer, we use simple invariants of a thresholded belief graph—connected components and independent cycles—as utilities for selecting the next structural hypothesis test.

We consider a setting where an upstream predictor, causal discovery procedure, mechanistic model, graph-based neural model, or LLM-based extractor produces candidate edges with confidence scores. We do not model this upstream system explicitly; instead, we use a controlled corruption-and-oracle protocol. Starting from a ground-truth graph $G^\star$, we remove true edges, inject spurious edges, and assign probabilities to candidate edges. This corrupted graph defines a probabilistic graph-valued belief state.

We summarize structural ambiguity using a thresholded undirected version of this belief graph. The number of connected components β_0 captures fragmentation or missing bridge relations, while the number of independent cycles β_1 captures unresolved alternative pathways or spurious cyclic structure. In the DCC condensation example (Figure 1), a missing bond-forming edge and high-confidence spurious edges create a corrupted belief graph with both erased mechanistic evidence and phantom alternatives.

The task is to sequentially select *costly structural tests* under a fixed budget. Each test is a simulated oracle query about edge existence. The oracle returns noisy binary outcomes, after which edge beliefs are updated by Bayes' rule and the thresholded graph is recomputed. The key question is: *which structural hypothesis should be tested next to most efficiently reduce global graph ambiguity per unit cost?*

To address this, we introduce **Topological Complexity Minimization (TCM)**, a cost-aware adaptive testing policy for structural epistemic uncertainty. TCM combines β_0 and β_1 into a topological complexity score and greedily selects tests that maximize expected reduction in this score, normalized by test cost. Expectations are estimated by Monte Carlo simulation over possible oracle outcomes. TCM acts as a decision layer on top of an existing belief representation and does not prescribe how beliefs are obtained.

We evaluate TCM on two complementary graph-valued testing domains: medium-scale causal graph benchmarks, where edges represent causal relations, and atom-level mechanistic graphs from `mech-USPTO-31k` (45), where edges represent electron-flow events. We compare against local uncertainty heuristics and structure-aware baselines across cycle-dominated, fragmentation-dominated, and mixed corruption regimes.

In summary, this work makes three contributions:

1. We formulate *adaptive structural hypothesis testing* for uncertain graph-valued ML systems, where candidate edges are dependent hypotheses and test selection must account for global graph structure.
2. We introduce TCM, a cost-aware adaptive testing policy based on a simple topological complexity measure using (β_0, β_1) on a thresholded belief graph.
3. We evaluate TCM on causal graph benchmarks and mechanistic graphs from `mech-USPTO-31k`, showing that topology-aware testing improves graph-level efficiency while preserving edge-level recovery, especially under missing bridges, fragmentation, and mixed structural ambiguity.

2. Topological Complexity Minimization as Adaptive Structural Hypothesis Testing

We now formalize **Topological Complexity Minimization (TCM)** as an adaptive testing policy for graph-valued hypotheses. The setting is a sequential structural hypothesis testing problem: the system maintains beliefs over candidate edges, adaptively chooses which edge or edge set to test next, observes a noisy binary outcome, and updates its belief state under a finite testing budget.

Graph-valued structural hypotheses. Let $G^\star = (V, E^\star)$ denote the unknown ground-truth relational structure. We maintain a candidate edge set E and a probabilistic belief graph $G = (V, E, W)$, where $W = \{p_{ij} \mid (i,j) \in E\}$ and $p_{ij} \in [0,1]$ is the current belief that edge (i,j) belongs

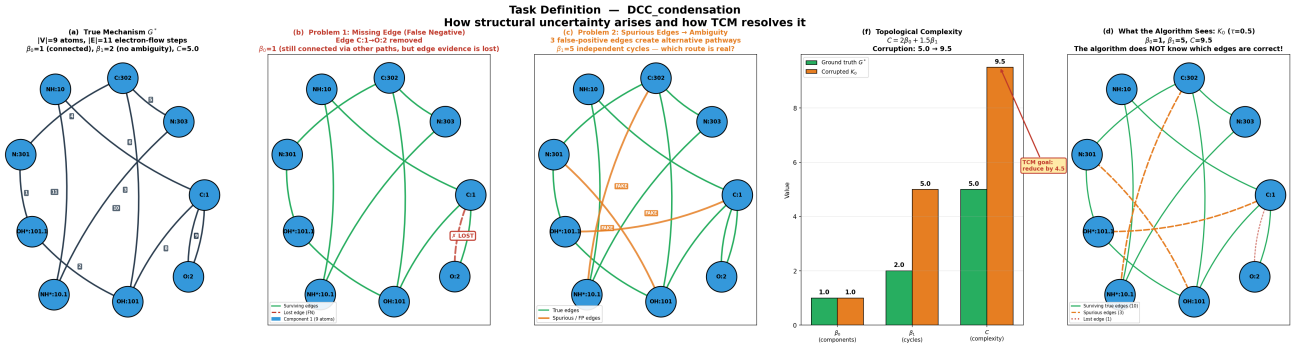


Figure 1. **Adaptive structural hypothesis testing on a real DCC condensation mechanism. (a)** Ground-truth graph $G^\star$ with atom-level nodes and electron-flow edges. **(b)** A false negative removes edge C:1→O:2, erasing direct evidence of the bond-forming step. **(c)** False-positive edges introduce spurious cycles (β_1=5), creating structural ambiguity. **(d)** The combined corrupted belief graph K_0 as seen by the algorithm, which cannot distinguish true from spurious edges.

to $E^\star$. Each candidate edge defines a binary structural hypothesis test,

$$H^0_{ij} : e_{ij} = 0 \qquad \text{versus} \qquad H^1_{ij} : e_{ij} = 1.$$

Unlike classical multiple testing over independent hypotheses, these edge hypotheses are structurally coupled: accepting or rejecting one edge can merge components, break cycles, or change the interpretation of many other candidate relations. The goal is therefore not merely to maximize local edge accuracy, but to adaptively select tests that reduce global uncertainty over the graph structure.

Topology as structural ambiguity. Given a confidence threshold τ, we form a thresholded graph K_τ containing all nodes in V and all candidate edges with posterior belief $p_{ij} > \tau$. For measuring global structural ambiguity, we use the undirected 1-skeleton of K_τ and summarize it by two Betti numbers:

$$\beta_0(K_\tau) = \#\mathrm{cc}(K_\tau), \qquad \beta_1(K_\tau) = m - n + \beta_0(K_\tau).$$

where $n = |V|$, m is the number of thresholded edges and cc represents connected components. Here β_0 measures fragmentation of the current structural explanation, while β_1 measures unresolved cyclic alternatives.

We define the topological complexity of the current structural hypothesis state as

$$C(K_\tau) = w_0\beta_0(K_\tau) + w_1\beta_1(K_\tau).$$

Large values indicate that the current graph-level hypothesis remains fragmented, cyclic, or both. The following proposition makes explicit how individual edge-level hypothesis decisions affect this graph-level objective.

Single-edge topology changes. The effect of a single structural test on the thresholded graph can be characterized locally. This clarifies why the two terms in $C(K_\tau)$ correspond to distinct testing modes.

Let $K = (V, E)$ be an undirected graph and let $K' = (V, E \cup \{e\})$ be obtained by adding a single edge $e = \{u, v\} \notin E$. Then exactly one of the following holds:

1. If u and v belong to different connected components of K, then

$$\beta_0(K') = \beta_0(K) - 1, \qquad \beta_1(K') = \beta_1(K).$$

2. If u and v belong to the same connected component of K, then

$$\beta_0(K') = \beta_0(K), \qquad \beta_1(K') = \beta_1(K) + 1.$$

Equivalently, removing a single edge can increase β_0 by at most one or decrease β_1 by at most one.

Proof. For any undirected graph, the first Betti number satisfies

$$\beta_1(K) = |E| - |V| + \beta_0(K).$$

Adding one edge increases $|E|$ by one while leaving $|V|$ unchanged. If the new edge connects two previously disconnected components, then β_0 decreases by one, so

$$\beta_1(K') = (|E| + 1) - |V| + (\beta_0(K) - 1) = \beta_1(K).$$

If the new edge lies within an existing connected component, then β_0 is unchanged, so

$$\beta_1(K') = (|E| + 1) - |V| + \beta_0(K) = \beta_1(K) + 1.$$

The removal statement follows by applying the same argument in reverse. □

Interpretation for adaptive testing. Proposition 2 shows that, at a fixed threshold, single-edge belief transitions fall into two graph-level testing modes. Edges whose posterior

transition connects two components act as *bridge-repair* hypotheses and primarily affect β_0. Edges whose transition removes or confirms within-component alternatives act as *cycle-resolution* hypotheses and primarily affect β_1. Thus, the TCM score decomposes the value of a structural test into expected component repair and expected cycle disambiguation, weighted by (w_0, w_1) and normalized by test cost.

Choice of weights. We score interventions with $C(K) = w_0\beta_0 + w_1\beta_1 + w_2\beta_2$ on the belief graph; in our 1-skeleton implementation $\beta_2 \equiv 0$, so (w_0, w_1) trade off reducing fragmentation against resolving cyclic ambiguity. We do not tune these weights on the benchmark, since doing so would mix the acquisition rule with dataset-specific fitting and could shift as the stratified mechanism mix grows with n. Instead, we fix a single interpretable prior: disconnected subgraphs are usually the first barrier to a coherent pathway, while cycle rank remains important but secondary. Concretely,

$$(w_0, w_1) = (2.0,\ 1.5),$$

slightly favoring β_0 over β_1. The same pair is used in *all* conditions so comparisons across regimes and baselines are not confounded by per-condition re-tuning informal sensitivity checks on smaller subsamples did not motivate a change under our full-protocol evaluation.

Adaptive test selection. At round t, the system considers a set of candidate tests/actions $a \in \mathcal{A}_t$. Each action may query one or more edge hypotheses and has cost $\mathrm{Cost}(a)$. TCM scores an action by its expected reduction in graph-level structural ambiguity,

$$\Delta C(a) = \mathbb{E}\left[C(K_t) - C(K_{t+1}) \mid a\right],$$

where the expectation is over possible noisy test outcomes under the current belief state. The selected test is

$$a_t^\star = \arg\max_{a \in \mathcal{A}_t} \frac{\Delta C(a)}{\mathrm{Cost}(a)}.$$

Thus, TCM is a cost-aware adaptive testing policy: it prioritizes the test expected to produce the largest topological simplification of the structural hypothesis space per unit cost. This parallels value-of-information criteria in Bayesian experimental design, but the utility is defined over graph structure rather than scalar predictive uncertainty.

Monte Carlo estimate of test value. For each candidate action a, we approximate $\Delta C(a)$ by Monte Carlo simulation. We compute the current complexity $C(K_t)$, draw $N = 5$ possible noisy test outcomes according to the oracle sensitivity and specificity, update queried edge beliefs under each sampled outcome, rebuild the thresholded graph, and average the resulting reductions $C(K_t) - C(K_{t+1})$. Since the acquisition score is used only to rank candidate tests, small N is sufficient in practice; Appendix D shows that increasing N yields similar rankings with higher runtime.

Bayesian belief updates after a test. A test returns a noisy binary outcome $o \in \{+, -\}$ indicating support for or against a queried edge hypothesis. If $e_{ij} \in \{0, 1\}$ denotes whether the edge is truly present, the posterior belief is updated as

$$p'_{ij} = \frac{P(o \mid e_{ij} = 1)p_{ij}}{P(o \mid e_{ij} = 1)p_{ij} + P(o \mid e_{ij} = 0)(1 - p_{ij})}.$$

The updated beliefs induce the next thresholded graph K_{t+1}, and the adaptive testing loop continues until the budget is exhausted.

Error control and discovery reporting. TCM is an adaptive *test-selection* policy, not a standalone multiple-testing correction procedure. Its role is to decide which dependent edge hypothesis should be queried next. When formal error control over reported discoveries is required, acquisition can be separated from reporting: TCM selects tests, while an optional e-value/eBH layer can be applied to the accumulated edge-level evidence to decide which hypotheses are declared discoveries. This reporting layer targets false discoveries in the declared edge set under the usual e-process assumptions; it does not certify recovery of the entire graph. We give one such optional construction and diagnostic experiment in Appendix E.

3. Experimental Evaluation

We evaluate *Topological Complexity Minimization* (TCM) as an adaptive structural hypothesis testing policy on two families of graph-valued testing problems: (i) medium-scale causal DAG benchmarks from `bnlearn` (73), and (ii) expert-annotated mechanistic reaction graphs from `mech-USPTO-31k`. The causal benchmarks represent standard structural hypothesis testing problems where candidate edges encode causal relations, while the mechanistic benchmarks instantiate the same acquisition setting on atom-level electron-flow graphs.

We study three uncertainty regimes: cycle-dominated corruption, fragmentation-dominated corruption, and mixed corruption. Although the underlying graphs are directed, all topological quantities are computed on the undirected 1-skeleton of the thresholded belief graph. We report both graph-level testing efficiency, measured by topological complexity reduction per unit cost, and F1.

Benchmark families. For the causal setting, we use the medium-scale `bnlearn` DAGs CHILD and INSURANCE (20–27 nodes, 25–52 edges). Since each benchmark provides a single canonical DAG, evaluation instances are

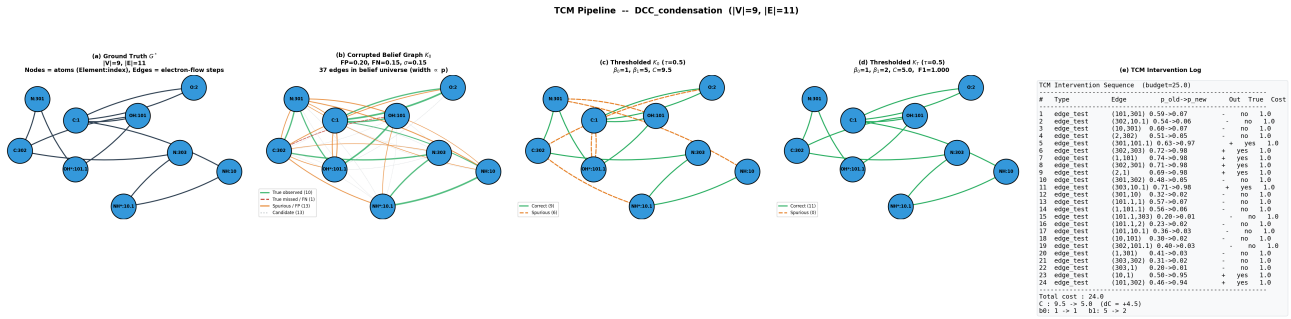


Figure 2. **Full TCM pipeline on a DCC condensation mechanism** ($|V|$=9, $|E|$=11). **(a)** Ground-truth graph $G^\star$. **(b)** Illustrative corrupted belief graph with elevated cyclic ambiguity (β_0=1, β_1=5, C=9.5). **(c)** Thresholding at τ=0.5 retains 9 correct and 6 spurious edges. **(d)** After 24 TCM-guided edge tests (budget = 25), spurious edges are removed: β_1 drops from 5 to 2, C returns to 5.0, and edge F1 reaches 1.000. **(e)** Intervention log showing the query sequence and probability updates.

generated by repeatedly corrupting the same ground-truth structure under different random seeds. For the mechanistic setting, we sample distinct reaction mechanisms from `mech-USPTO-31k`, yielding structurally diverse graph instances.

Graph construction from mechanistic steps. Each reaction in `mech-USPTO-31k` is annotated with elementary mechanistic steps describing electron flow between atom-mapped positions. We collect all referenced atom indices as nodes in $V^\star$ and convert each mechanistic step into directed source–target edges between the corresponding atoms, producing a directed atom-level electron-flow graph $G^\star$.

Structured corruption regimes. Corruption is controlled by base false-positive and false-negative rates (FP, FN) shared across regimes, but *where* errors concentrate is regime-dependent rather than i.i.d. uniform on all edges. **Cycle-dominated corruption** upweights false positives among pairs that lie within the same weakly connected region of the current noisy graph, encouraging spurious *alternative routes* and inflating β_1 while largely preserving connectivity (β_0). **Fragmentation-dominated corruption** upweights false negatives on connectivity-critical (bridge-like) true edges, encouraging *disconnecting* merges of coherent substructures and inflating β_0 with relatively modest additional cyclic ambiguity. **Mixed corruption** combines both mechanisms, yielding belief graphs that can be simultaneously fragmented and cycle-ambiguous.

Belief initialization (corrupted world model). Starting from a directed ground-truth graph $G^\star$, we obtain an uncertain initial belief graph by applying the regime-specific deletion/insertion process, then assigning Beta-like priors on edges. Observed true edges that remain present in the corrupted structural summary receive high-confidence beliefs $p_{ij} = 0.8 + \varepsilon$, with $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, clipped to $[0.5, 0.95]$ for stability. True edges targeted as false negatives remain in the candidate universe with weaker support $p_{ij} \sim \mathrm{Uniform}(0.2, 0.5)$. Injected false-positive edges receive intermediate beliefs $p_{ij} \sim \mathrm{Uniform}(0.3, 0.6)$. To model uncertainty over plausible but unverified relations, we additionally instantiate a *candidate edge universe*: a random subset of remaining non-edges is promoted to low prior support $p_{ij} \sim \mathrm{Uniform}(0.1, 0.3)$, with sampling intensity controlled by a `candidate_rate` parameter (implementation default 0.25; values in the 0.25–0.30 range are used in some configurations). Unless otherwise stated, we use $\mathrm{FP} = 0.2$, $\mathrm{FN} = 0.15$, and $\sigma = 0.15$ across benchmarks.

Interventions, oracle, and costs. Strategies select among three intervention families: `edge_test`, which queries whether an edge exists in $G^\star$; `bridge_search`, which probes candidate edges spanning disconnected components; and `cycle_disamb`, which targets cycle-inducing edges in the thresholded graph.

Oracle outcomes are noisy with sensitivity $\eta = 0.95$ and specificity $\xi = 0.95$:

$$P(o{=}+ \mid e_{ij}{=}1) = \eta, \qquad P(o{=}+ \mid e_{ij}{=}0) = 1 - \xi.$$

Queried beliefs are updated via Bayes' rule and clipped to $[0.01, 0.99]$. We use the cost hierarchy: `bridge_search` = 0.3, `edge_test` = 1.0, and `cycle_disamb` = 2.0.

Testing interpretation. Each query is a noisy test of $H^0_{ij} : e_{ij} = 0$ against $H^1_{ij} : e_{ij} = 1$. The oracle specificity ξ determines the type-I error probability $\alpha = 1 - \xi$, while sensitivity η determines power $1 - \beta = \eta$. Thus, the belief update can be interpreted as sequential Bayesian testing under a known test error model.

Action instantiation. `edge_test` targets uncertain edges with $0.2 < p_{ij} < 0.8$. `cycle_disamb` targets non-tree edges with $0.5 < p_{ij} < 0.75$. `bridge_search`

targets candidate edges connecting different components with $0.1 < p_{ij} < 0.7$. We cap `cycle_disamb` and `bridge_search` candidates at 3 and 5 per iteration, respectively.

Metrics. Structural ambiguity is summarized by

$$C(K_t) = 2.0\,\beta_0(K_t) + 1.5\,\beta_1(K_t),$$

computed on the thresholded graph with $\tau = 0.5$. We report total complexity reduction $\Delta C = C(K_0) - C(K_T)$, structural efficiency $\Delta C/\mathrm{Cost}$, and final (β_0, β_1) values.

Edge recovery is evaluated by thresholding final beliefs at $\tau = 0.5$ and comparing predicted edges to $E^\star$.

Baselines. We compare TCM to seven baselines spanning probability-centric uncertainty heuristics, lightweight structure-aware proxies, and a random control. Probability-centric baselines include marginal edge entropy (Entropy), Bayesian Active Learning by Disagreement (BALD) (74), and pairwise mutual information (MI) (50), each prioritizing local uncertainty reduction. Structure-aware baselines include CYCLERANK, a cycle-focused heuristic inspired by graph-theoretic cycle elimination, as well as DegEntropy (degree-distribution entropy) (75) and Spectral (Laplacian spectral entropy) (76), which summarize coarse global graph statistics without explicitly optimizing structural disambiguation. Random selects interventions uniformly. All methods have access to the same candidate interventions, oracle model, and costs.

Regime-wise structural testing. Our main evaluation compares all strategies across three structured uncertainty regimes: cycle-dominated, fragmentation-dominated, and mixed corruption. This tests whether TCM adapts to different forms of graph-valued uncertainty: spurious cycles, missing bridges, or both. We run the same protocol on both `bnlearn` Bayesian network DAGs and mechanistic graphs from `mech-USPTO-31k`.

`bnlearn` DAG setting. We evaluate on the CHILD and INSURANCE Bayesian network DAGs from `bnlearn`. Since each dataset provides a single canonical ground-truth DAG, repeated instances correspond to independent corrupted belief states generated under different random seeds.

Mechanistic graph setting. We evaluate on distinct ground-truth mechanisms sampled from `mech-USPTO-31k`. Unlike the `bnlearn` setting, this benchmark tests whether the same adaptive testing policy generalizes across many different scientific graph structures.

Hyperparameter and runtime sensitivity. We ablate the topology weights (w_0, w_1), threshold τ, and Monte Carlo sample count N on representative causal and mechanistic subsets. We also report runtime per iteration to quantify the computational cost of Monte Carlo action scoring relative to cheaper uncertainty-based baselines. Results can be found in Appendix D.

4. Results

We compare *Topological Complexity Minimization* (TCM) to seven baselines across two benchmark families: medium-scale `bnlearn` DAGs (CHILD, INSURANCE) and mechanistic reaction graphs from `mech-USPTO-31k`. Results are organized by structural uncertainty regime: cycle-dominated, fragmentation-dominated, and mixed corruption. Unless otherwise stated, we report averages over graph instances and random seeds.

Regime-wise performance on mechanistic graphs. On 200 stratified mechanisms from `mech-USPTO-31k`, TCM achieves the highest structural efficiency in all three corruption regimes (Table 1). In the cycle-dominated regime, TCM is nearly tied with the specialized CYCLERANK baseline (0.512 versus 0.506), as expected when the dominant ambiguity is cyclic (Figure 4). The gap becomes larger in the fragmentation regime (0.576 versus 0.452), where missing connectivity-critical edges make bridge recovery important. The highest absolute mechanistic efficiency appears in the mixed regime, where TCM reaches 0.830 compared with 0.715 for CYCLERANK and 0.537 for Random. Across all regimes, TCM maintains edge-level F1 comparable to the best baselines, indicating that its higher efficiency is not obtained by sacrificing edge recovery. Figure 7 shows the corresponding per-regime mechanistic distributions.

`bnlearn` structural hypothesis testing. On the medium-scale `bnlearn` DAGs, TCM is again the most efficient strategy across the three structured regimes (Table 1). In cycle-dominated corruption, TCM performs similarly to cycle-focused baselines: TCM achieves efficiency 1.593, while both CYCLERANK and SPECTRAL achieve 1.500. In fragmentation-dominated corruption, TCM improves more clearly over CYCLERANK (1.471 versus 1.062), consistent with the role of the β_0 term in prioritizing bridge hypotheses. In mixed corruption, TCM obtains the largest gap, with efficiency 2.049 compared with 1.406 for CYCLERANK and 0.878 for Entropy/BALD. This suggests that topology-aware testing is especially useful when the belief graph contains both missing bridges and spurious alternative routes. Figure 8 shows the same regime-separated visualization for the two-graph causal pool (CHILD, INSURANCE); qualitatively, the efficiency ordering matches Table 1, while absolute scales differ

Table 1. **Regime-wise performance across mechanistic and causal benchmarks.** We report mean structural efficiency $\Delta C/\mathrm{Cost}$ and mean edge F1. Best values within each benchmark/regime are bolded, and second-best values are underlined.

	Mechanistic: mech-USPTO-31k (n=200)						**`bnlearn` : Child + Insurance**					
	Cycle		**Fragment**		**Mixed**		**Cycle**		**Fragment**		**Mixed**	
Strategy	Eff	F1	Eff	F1	Eff	F1	Eff	F1	Eff	F1	Eff	F1
TCM	**0.512**	0.888	**0.576**	**0.959**	**0.830**	**0.906**	**1.593**	**0.464**	**1.471**	0.928	**2.049**	**0.759**
CycleRank	0.506	0.890	0.452	0.955	0.715	0.905	1.500	0.460	1.062	0.860	1.406	0.690
Spectral	0.474	0.886	0.333	0.917	0.481	0.822	1.500	0.460	0.656	0.861	1.031	0.698
DegEntropy	0.256	0.851	0.361	0.943	0.451	0.845	1.344	0.459	1.032	0.916	0.414	0.663
MI	0.210	0.841	0.291	0.896	0.415	0.853	0.305	0.421	0.909	0.889	0.808	0.675
Random	0.155	0.855	0.410	0.943	0.537	0.875	0.094	0.441	0.281	0.847	0.492	0.669
Entropy	0.108	**0.903**	0.252	0.925	0.406	0.903	0.579	0.461	0.470	**0.940**	0.878	0.713
BALD	0.108	**0.903**	0.252	0.925	0.406	0.903	0.579	0.461	0.470	**0.940**	0.878	0.713

because causal graphs induce different initial complexity and candidate-edge geometry.

Structural efficiency versus edge-level recovery. A consistent pattern across both benchmark families is that local uncertainty baselines can achieve strong edge-level recovery while remaining inefficient at graph-level disambiguation. For example, on `bnlearn` fragmentation regimes, Entropy and BALD achieve the highest mean F1 (0.940) but much lower structural efficiency (0.470) than TCM (1.471). Similarly, on mechanistic mixed corruption, Entropy and BALD preserve high F1 (0.903) but achieve roughly half of TCM's efficiency (0.406 versus 0.830). These results support the central distinction between local edge prediction and adaptive structural hypothesis testing: the most uncertain edge is not necessarily the most structurally informative test.

Effect of structural regime. The advantage of TCM depends on the type of uncertainty. When corruption is cycle-dominated, specialized cycle-aware methods are strong and TCM mostly matches them. When corruption fragments the graph or mixes fragmentation with spurious cycles, TCM more clearly outperforms both cycle-only and uncertainty-only baselines. This regime dependence is desirable: it shows that TCM behaves like a general structural testing policy rather than a method tuned only to cycle elimination.

5. Discussion

Adaptive structural testing is different from local edge recovery. Across both benchmark families, TCM consistently achieves the highest structural efficiency, even when local uncertainty baselines achieve comparable or better edge-level F1. This distinction is central to our setting. In graph-valued hypothesis testing, the most uncertain edge is not necessarily the most informative test: resolving one edge may reconnect components, collapse alternative routes, or leave the global hypothesis structure essentially unchanged. The results in Table 1 show that TCM prioritizes tests that reduce graph-level ambiguity per unit cost, rather than simply maximizing local edge classification accuracy.

Cycle-focused heuristics are strong, but specialized. In cycle-dominated regimes, CycleRank and other cycle-focused structure proxies are competitive baselines, and TCM nearly matches or slightly exceeds them on efficiency while maintaining similar edge F1. This is expected: when the main structural ambiguity consists of spurious alternative paths, heuristics that directly target cyclic structure can match much of the signal that TCM pursues through expected ΔC. The important difference appears outside this special case. In fragmentation and mixed regimes, TCM outperforms CycleRank because the objective accounts for both disconnected components (β_0) and unresolved cycles (β_1), whereas CycleRank primarily rewards cycle reduction. Thus, the empirical story is not that TCM dominates every specialized heuristic in every regime, but that it provides a unified testing policy across different forms of structural uncertainty.

The β_0 term matters in fragmentation-heavy settings. Large gains occur when corruption removes connectivity-critical edges, especially in fragmentation-heavy settings. In these settings, entropy-type uncertainty scores and BALD coincide numerically under our intervention parameterization (Table 1), yet they can still recover many individual edges while spending budget on locally uncertain hypotheses that do not reconnect the graph. TCM instead assigns high value to tests whose outcomes are expected to reduce fragmentation. This explains the strong gains in the fragmentation regimes of both the mechanistic and `bnlearn` evaluations. Figure 3 pools all mechanistic regime runs into a single summary (efficiency, edge F1, best-overall win rate, and an efficiency–F1 scatter), making the recurring trade-off explicit: probability-centric baselines can remain competitive on edge F1 while using budget less effectively on global structural simplification.

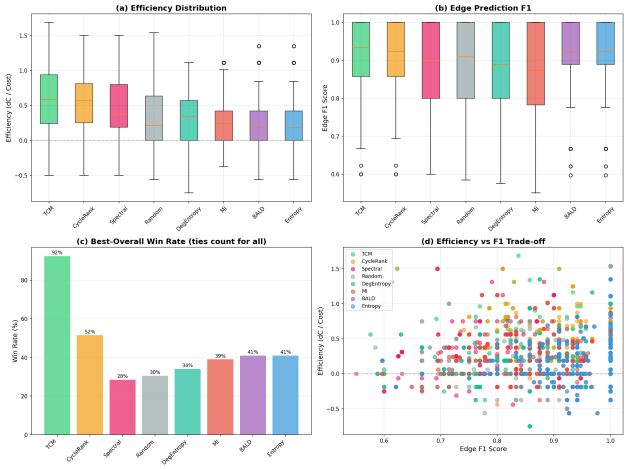


Figure 3. **Mechanistic results pooled across regimes.** Same four-panel summary as Figure 7, but aggregating cycle, fragment, and mixed runs together, emphasizing how often each method sits on the empirical efficiency–F1 frontier under the shared protocol.

Mixed regimes expose the value of cost-aware topology. Mixed corruption combines missing bridges with spurious alternative routes. These settings are challenging because a useful policy must trade off component repair, cycle disambiguation, local uncertainty, and test cost. TCM performs best in the aggregate mixed results, suggesting that the cost-normalized topological objective captures this trade-off better than purely local uncertainty or cycle-only heuristics. In these mixed settings, CycleRank remains the strongest non-TCM efficiency baseline, but simpler heuristics can still be competitive on some instances; for example, Random reaches 0.537 efficiency on mechanistic mixed corruption, ahead of several uncertainty-based baselines. The mixed setting is also where fixed topology weights may be most limiting: future work could adapt (w_0, w_1) online based on the current belief graph or learn the acquisition utility from previous testing episodes.

Mechanistic and causal benchmarks play complementary roles. The `bnlearn` experiments instantiate a standard structural hypothesis testing setting where candidate edges encode directed probabilistic dependencies. Because each `bnlearn` dataset provides one canonical DAG, these experiments test robustness across corrupted belief states of the same underlying graph. In contrast, `mech-USPTO-31k` provides many distinct ground-truth mechanisms, testing whether the same decision rule generalizes across diverse scientific graph structures. Agreement across these settings suggests that TCM is not tied to a single chemistry-specific benchmark, while the mechanistic results provide a realistic scientific application where structural ambiguity is naturally graph-valued.

Computation trades off against intervention efficiency. TCM is more expensive than Entropy or BALD because it estimates expected topological reduction by Monte Carlo simulation. However, the additional computation is modest at the graph sizes considered here and is often negligible relative to the cost of real scientific interventions, expert queries, or simulations. The main scalability challenge is not the topological summary itself, but repeatedly scoring candidate actions. Caching single-edge topology updates, pruning the action set, or using learned proposal policies could reduce runtime without changing the testing objective.

6. Conclusion

We introduced *Topological Complexity Minimization* (TCM), a cost-aware policy for adaptive testing of dependent graph-valued hypotheses. TCM treats edge tests as structurally coupled: resolving one edge can reconnect components, collapse spurious alternatives, or leave the global hypothesis state unchanged. It captures this dependence with a lightweight topological utility based on connected components (β_0) and independent cycles (β_1).

Across causal DAG benchmarks and mechanistic graphs from `mech-USPTO-31k`, TCM achieves the strongest graph-level testing efficiency across cycle-dominated, fragmentation-dominated, and mixed corruption regimes. These results show that adaptive structural hypothesis testing is not equivalent to selecting the most uncertain edge: local uncertainty methods can recover many edges while spending budget on tests that do little to simplify the global graph. Overall, TCM provides an interpretable utility for hypothesis testing under adaptivity and structural dependence. Future work should develop tighter error-controlled acquisition rules, filtration-based objectives, correlated oracle models, and real experimental or LLM-in-the-loop testing settings. Code will be released upon acceptance.

References

[1] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, UK, 2 edition, 2009.

[2] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, Cambridge, MA, 2 edition, 2000.

[3] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[4] Jie Zhou, Ganqu Cui, Shengding Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

[5] Thérèse Gaudelet et al. Utilizing graph machine learning within drug discovery and development. *Briefings in Bioinformatics*, 22(4):bbaa430, 2021.

[6] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1050–1059, 2016.

[7] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110:457–506, 2021.

[8] Moloud Abdar, Farhad Pourpanah, Saeid Hussain, Dana Rezazadegan, Lizhe Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021.

[9] Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 30, 2017.

[10] Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *Advances in Neural Information Processing Systems*, 33:4697–4708, 2020.

[11] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

[12] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2796–2804, 2018.

[13] Yaniv Ovadia et al. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[14] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *Advances in Neural Information Processing Systems*, 31, 2018.

[15] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

[16] Stephan Bongers, Jonas Peters, Joris M. Mooij, and Andrew Markham. Foundations of structural causal models with cycles and latent variables. *Annals of Statistics*, 49(5):2885–2915, 2021.

[17] Jakob Runge, Sebastian Bathiany, Erik Bollt, et al. Inferring causation from time series in earth system sciences. *Nature Communications*, 10:2553, 2019.

[18] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[19] Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. Integrating physics-based modeling with machine learning: A survey. *ACM Computing Surveys*, 55(3):1–34, 2020.

[20] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1263–1272, 2017.

[21] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018.

[22] David Heckerman, Dan Geiger, and David M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

[23] David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002.

[24] Xun Zheng, Bryon Aragam, Pradeep K. Ravikumar, and Eric P. Xing. DAGs with NO TEARS: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[25] Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. Masked gradient-based causal structure learning. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pages 370–378. SIAM, 2022.

[26] Hang Yu, Songwei Wu, and Justin Dauwels. Efficient variational bayes learning of graphical models with smooth structural changes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):475–488, 2022.

[27] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021.

[28] Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. Learning discrete structures for graph neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1972–1982. PMLR, 2019.

[29] Marwin HS Segler, Mike Preuss, and Mark P Waller. Planning chemical syntheses with deep neural networks and symbolic ai. *Nature*, 555(7698):604–610, 2018.

[30] James Law, Mire Zloh, et al. Machine learning in chemical reaction space. *Chemical Reviews*, 119(16):9871–9908, 2019.

[31] Connor W. Coley, Jin Wengong, , Luke Rogers, Regina Barzilay, William H. Green, and Klavs F. Jensen. Computer-assisted retrosynthesis based on molecular similarity. *ACS Central Science*, 3(12):1237–1245, 2017.

[32] Gisbert Schneider and Uli Fechner. Computer-based de novo design of drug-like molecules. *Nature Reviews Drug Discovery*, 4(8):649–663, 2005.

[33] Connor W. Coley, Wengong Jin, Luke Rogers, Timothy F. Jamison, Tommi S. Jaakkola, William H. Green, Regina Barzilay, and Klavs F. Jensen. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chem. Sci.*, 10:370–377, 2019.

[34] John Bradshaw, Brooks Paige, Matt J. Kusner, Marwin Segler, and José Miguel Hernández-Lobato. A model to search for synthesizable molecules. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[35] Yiyuan Xu, Kexin Lin, Sheng Wang, Lei Wang, Chao Cai, Chengfeng Song, Luhua Lai, and Jianfeng Pei. Deep learning for molecular generation. *Future Medicinal Chemistry*, 11(6):567–597, 2019. Epub 2019-01-30.

[36] Philippe Schwaller, Théophile Gaudin, Dávid Lányi, Costas Bekas, and Teodoro Laino. "found in translation": predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models. *Chem. Sci.*, 9:6091–6098, 2018.

[37] Philippe Schwaller, Teodoro Laino, Thierry Gaudin, Peter Bolgar, Christopher A. Hunter, Costas Bekas, and Alpha A. Lee. Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS Central Science*, 5(9):1572–1583, 2019.

[38] Philippe Schwaller, Rocco Petraglia, Valerio Zullo, and et al. Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy. *Chemical Science*, 11(12):3316–3325, 2020. Published 2020-03-03.

[39] Jonathan M. Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Juan R. Cubillos-Ruiz, Nicole M. Donghia, Craig R. MacNair, Samuel French, Lauren A. Carfrae, Zachary Bloom-Ackermann, et al. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702, 2020.

[40] Paul M Zimmerman. Automated discovery of chemically reasonable elementary reaction steps. *Journal of Computational Chemistry*, 34(16):1385–1392, 2013.

[41] Matthew A. Kayala and Pierre Baldi. Learning to predict chemical reaction outcomes: Challenges and opportunities. In *Proceedings of the 2011 International Conference on Machine Learning and Applications*, pages 1–8, 2011.

[42] Matthew A. Kayala, Céline-A. Azencott, Judith H. Chen, and Pierre Baldi. Predicting organic chemistry reactions with neural networks. *Journal of Chemical Information and Modeling*, 51(9):2209–2222, 2011.

[43] David Fooshee, Alexander Mood, Eyal Gutman, Meric Tavakoli, Alexander Urban, Fang Liu, Wenlong Hu, and Pierre Baldi. Deep learning for chemical reaction prediction. *Molecular Systems Design & Engineering*, 3(3):442–452, 2018.

[44] Kjell Jorner, Tore Brinck, Per-Ola Norrby, and David Buttar. Machine learning meets mechanistic modelling for accurate prediction of experimental activation energies. *Chem. Sci.*, 12:1163–1175, 2021.

[45] Shuan Chen, Ramil Babazade, Taewan Kim, Sunkyu Han, and Yousung Jung. A large-scale reaction dataset of mechanistic pathways of organic reactions. *Scientific Data*, 11(1):863, 2024.

[46] Dennis V. Lindley. On a measure of the information provided by an experiment. *Annals of Mathematical Statistics*, 27(4):986–1005, 1956.

[47] Valerii V. Fedorov. *Theory of Optimal Experiments*. Academic Press, New York, 1972.

[48] José M. Bernardo. Expected information as expected utility. *Annals of Statistics*, 7(3):686–690, 1979.

[49] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995.

[50] David JC MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

[51] Burr Settles. Active learning literature survey. Technical Report 1648, University of Wisconsin–Madison, 2009.

[52] Peter I. Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[53] Lu Lu and Christine Anderson-Cook. Strategies for sequential design of experiments and augmentation. *Quality and Reliability Engineering International*, 37:1740–1757, 12 2020.

[54] Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. volume 42, pages 427–486, 2011.

[55] Peter I. Frazier, Warren B. Powell, and Savas Dayanik. *A Knowledge-Gradient Policy for Sequential Information Collection*, volume 47. 2008.

[56] Kirthevasan Kandasamy, Willie Neiswanger, Jeff Schneider, Barnabas Poczos, and Eric Xing. Neural architecture search with bayesian optimisation and optimal transport. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[57] Shibo Li, Wei Xing, Robert Kirby, and Shandian Zhe. Multi-fidelity bayesian optimization via deep neural networks. volume 33, pages 8521–8531, 2020.

[58] Justin S. Smith, Ben Nebgen, Nicholas Lubbers, Olexandr Isayev, and Adrian E. Roitberg. Less is more: Sampling chemical space with active learning. *Journal of Chemical Theory and Computation*, 14(11):6341–6351, 2018.

[59] Benjamin P. MacLeod, Fiona G. L. Parlane, Tess D. Morrissey, Florian Häse, Loïc M. Roch, Kevin E. Dettelbach, Giovanna J. Ng, Haiyuan Situ, Rui Zhang, Margaret S. Elliott, Joshua C. Haley, Rodrigo Moreira, Luke P. E. Yunker, J. Alexander love, Jason E. Hein, and Curtis P. Berlinguette. Self-driving laboratory for accelerated discovery of thin-film materials. *Science Advances*, 6(20):eabb2219, 2020.

[60] Rafael Gómez-Bombarelli, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, David Duvenaud, Dougal Maclaurin, Martin A. Blood-Forsythe, Heejoo Chae, Markus Einzinger, Dong-Gwang Ha, Tian Wu, Georgios Markopoulos, Seongkuk Jeon, Peter Kang, Hiroaki Miyazaki, Masaru Numata, Seokjoon Kim, Wilson Huang, Seonghoon Hong, Marc A. Baldo, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic discovery of materials and catalysts by quantum mechanics and machine learning. *ACS Central Science*, 4(2):268–276, 2018.

[61] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

[62] Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. Applied Mathematics. American Mathematical Society, 2010.

[63] Robert Ghrist. Barcodes: The persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.

[64] Nina Otter, Mason A. Porter, Ulrike Tillmann, Peter Grindrod, and Heather A. Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6(1):17, 2017.

[65] Yuichi Hiraoka, Tomoyuki Nakamura, Akihiko Hirata, Edgar G. Escolar, Kanako Matsue, and Yasumasa Nishiura. Hierarchical structures of amorphous solids characterized by persistent homology. *Proceedings of the National Academy of Sciences*, 113(26):7035–7040, 2016.

[66] Z. Dong, J. Pu, and H. Lin. Multiscale persistent topological descriptor for porous structure retrieval. *Computer Aided Geometric Design*, 88:102004, 2021.

[67] Y. Lee, S. D. Barthel, P. Dłotko, S. M. Moosavi, K. Hess, and B. Smit. Quantifying similarity of pore-geometry in nanoporous materials. *Nature Communications*, 8:15300, 2017.

[68] Chad Giusti, Eva Pastalkova, Carina Curto, and Vladimir Itskov. Clique topology reveals intrinsic geometric structure in neural correlations. *Proceedings of the National Academy of Sciences*, 112(44):13455–13460, 2015.

[69] Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. *Advances in Neural Information Processing Systems*, 30, 2017.

[70] Felix Hensel, Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. A survey of topological machine learning methods. *Frontiers in Artificial Intelligence*, 4:681108, 2021.

[71] James R Clough, Ilkay Oksuz, Nicholas Byrne, Julia A Schnabel, and Andrew P King. Explicit topological priors for deep-learning based image segmentation using persistent homology. pages 16–28, 2019.

[72] Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. Topological autoencoders. *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[73] Marco Scutari. Learning bayesian networks with the bnlearn R package. *Journal of Statistical Software*, 35(3):1–22, 2010.

[74] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[75] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016.

[76] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[77] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.

[78] Matthew Kahle. Topology of random simplicial complexes: A survey. *AMS Contemporary Mathematics*, 620:201–222, 2014.

[79] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

[80] Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J. Guibas, and Steve Y. Oudot. Proximity of persistence modules and their diagrams. In *Proceedings of the Twenty-Fifth Annual Symposium on Computational Geometry (SoCG)*, pages 237–246. ACM, 2009.

[81] Vladimir Vovk and Ruodu Wang. E-values: Calibration, combination and applications. *The Annals of Statistics*, 49(3):1736–1754, 2021.

[82] Ruodu Wang and Aaditya Ramdas. False discovery rate control with e-values. *Journal of the Royal Statistical Society: Series B*, 84(3):822–852, 2022.

A. TCM Workflow

Algorithm 1 TCM decision loop (fixed-threshold 1-skeleton implementation)

Input: $G^\star = (V^\star, E^\star)$, B, τ, (w_0, w_1), (η, ξ), N
Output: $(a_t)_{t=0}^{T-1}$, (p_{ij})
Init: corrupt $G^\star$ to obtain beliefs $\{p_{ij}\}$; $b \leftarrow 0$
while $b < B$ and candidates remain **do**
  Construct $K_t = (V, E_t)$ with $E_t = \{(i,j) : p_{ij} > \tau\}$
  Compute $\beta_0(K_t)$; $\beta_1(K_t) = |E_t| - |V| + \beta_0(K_t)$
  $C(K_t) = w_0\beta_0(K_t) + w_1\beta_1(K_t)$
  Instantiate $\mathcal{A}_t$ from K_t and beliefs
  for all $a \in \mathcal{A}_t$ **do**
    Estimate $\Delta C(a)$ via N MC samples
    for $s = 1$ to N **do**
      Sample oracle outcome $o_s \in \{+, -\}$
      Update queried edge belief to get p'_{ij}
      Recompute $C(K_{t+1}^{(s)})$
    end for
    $S(a) \leftarrow \Delta C(a)/\mathrm{Cost}(a)$
  end for
  Choose $a_t^\star \leftarrow \arg\max_{a \in \mathcal{A}_t} S(a)$
  Query oracle; update beliefs; $b \leftarrow b + \mathrm{Cost}(a_t^\star)$
end while

B. General Topological Formulation

Purpose and scope. The main paper intentionally uses a *fixed-threshold, 1-skeleton* instantiation of TCM because it is (i) inexpensive to compute in an iterative decision loop, and (ii) directly interpretable in terms of mechanistic fragmentation (components) and unresolved alternatives (cycles). This appendix records a more general formulation that places TCM within the standard filtered-complex and persistent-homology framework (62; 77; 63; 64). We emphasize that **persistent homology is not required for, and is not used in, any experiments reported in the main paper**; it is included only to clarify how the same acquisition principle can extend beyond graphs and a single threshold.

Filtration from edge confidences (general form). Let $G = (V, E, W)$ be a belief graph with edge confidences $p_{ij} \in [0,1]$. In the most general form of TCM, we view a belief state as a *filtered simplicial complex* (62; 64). Let K be a simplicial complex built from the undirected 1-skeleton of G (e.g., a clique/flag complex) (78), and let $f : K \to [0,1]$ assign a *birth time* to each simplex. Using a superlevel filtration induced by edge confidence,

$$t \in [0,1] : \quad \sigma \in K_t \iff f(\sigma) \geq t, \qquad f(\{i,j\}) = \bar{p}_{ij},$$

and extending to higher simplices by the standard lower-star construction (62),

$$f(\sigma) = \min\{f(\tau) : \tau \subset \sigma,\ \dim \tau = 1\},$$

yields a nested sequence $K_1 \subseteq K_{t_2} \subseteq \cdots \subseteq K_0$. This construction supports the usual persistence stability guarantees under perturbations of f (79; 80).

Homology and Betti numbers. Over a field $\mathbb{F}$ (we use $\mathbb{F}_2$), the chain groups $C_k(K;\mathbb{F})$ are free vector spaces generated by k-simplices, with boundary maps $\partial_k : C_k \to C_{k-1}$ (61). The k-th homology group is

$$H_k(K;\mathbb{F}) = \ker(\partial_k)/\mathrm{im}(\partial_{k+1}),$$
$$\beta_k(K) = \mathrm{rank}\, H_k(K;\mathbb{F}).$$

Equivalently, using boundary matrices D_k,

$$\beta_k(K) = |S_k| - \mathrm{rank}(D_k) - \mathrm{rank}(D_{k+1}),$$

where S_k denotes the set of k-simplices.

Topological complexity objective and action value (general form). Given a filtration level t, define a weighted topological complexity

$$C(K_t) = \sum_{k \geq 0} w_k \beta_k(K_t), \qquad w_k \geq 0,$$

and score candidate interventions by expected complexity reduction

$$\Delta C(a) = \mathbb{E}[C(K_t) - C(K_{t+1}) \mid a],$$
$$a_t^\star = \arg\max_a \frac{\Delta C(a)}{\mathrm{Cost}(a)}.$$

In this general view, interventions update beliefs (and thus the filtration), and TCM prioritizes actions that are expected to simplify the induced topology.

Implemented specialization used in this paper. All experiments in the main paper use a fixed-threshold 1-skeleton specialization: we compute (β_0, β_1) on the thresholded undirected graph at $\tau = 0.5$ (with β_1 via the cycle-rank identity) and set $\beta_k = 0$ for $k \geq 2$. This choice isolates the decision-layer contribution of TCM while keeping per-iteration computation practical; exploring filtration-based objectives and higher-order summaries is left for future work.

C. Implementation Details

Graph support and undirected topology. All mechanisms are directed graphs, but all topological quantities are computed on the undirected 1-skeleton obtained by symmetrizing queried edges. The node set V is fixed to $V^\star$ for the entire episode (we do not drop isolated nodes).

Candidate edge universe. Each mechanism maintains a fixed candidate edge set $\mathcal{E}_{\text{cand}}$ consisting of (i) all edges observed after corruption and (ii) additional low-prior candidates sampled uniformly from non-edges of $G^\star$ (used only to enable `bridge_search`). Beliefs are stored only for edges in $\mathcal{E}_{\text{cand}}$.

Numeric stability. Beliefs are clipped after each update to $p_{ij} \in [0.01, 0.99]$. We use a fixed threshold $\tau = 0.5$ for topology and for edge-level prediction.

Action ranking and tie-breaking. Actions are ranked by $S(a) = \Delta C(a)/\text{Cost}(a)$. Ties are broken deterministically by preferring `bridge_search` over `edge_test` over `cycle_disamb`, then by lexicographic edge id.

Per-iteration runtime. Each iteration computes connected components in $O(|V| + |E_t|)$ and evaluates $|\mathcal{A}_t|$ actions with N Monte Carlo samples; the dominant cost is $O(N\,|\mathcal{A}_t|\,(|V| + |E_t|))$ in the current implementation.

Why not learned acquisition baselines? We intentionally focus on non-learned baselines because our goal is to isolate the effect of the *decision-layer acquisition objective* under a controlled corruption-and-oracle protocol. Introducing a learned policy (e.g., RL or GNN-based action selection) would entangle acquisition performance with training data, architectural choices, reward shaping, and hyperparameter tuning, making it unclear whether improvements arise from learning capacity or from the structural objective itself.

It can also be noted that learned acquisition policies typically require many training episodes across mechanisms and can overfit to the specific corruption and oracle model used for evaluation. In contrast, TCM is *training-free* and designed as a lightweight drop-in criterion that can operate downstream of any hypothesis generator. To partially cover the space of structure-aware alternatives without learning, we include global structural baselines such as cycle-targeting heuristics and spectral graph entropy. Comparison to learned structural active-learning policies is an important direction for future work, and our oracle-based setup makes it straightforward to add such baselines in a follow-up study.

D. Hyperparameter and Monte Carlo runtime sensitivity

We isolate the knobs of the TCM objective that appear in the main text: topology weights (w_0, w_1) on merge- versus loop-seeking terms (with w_2=1 fixed), the belief threshold τ used both to binarize the topology score and to threshold edge posteriors, and the Monte Carlo sample count N_{MC} used when estimating expected ΔC per candidate intervention. On the same n=25 stratified subsample of

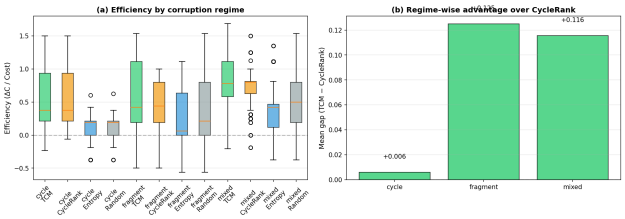


Figure 4. **Mechanistic regime comparison (subset of baselines).** (a) Efficiency ($\Delta C/\text{Cost}$) across corruption regimes. (b) Mean efficiency gap between TCM and CycleRank (TCM − CycleRank). Full eight-strategy breakdowns appear in Figures 7–3.

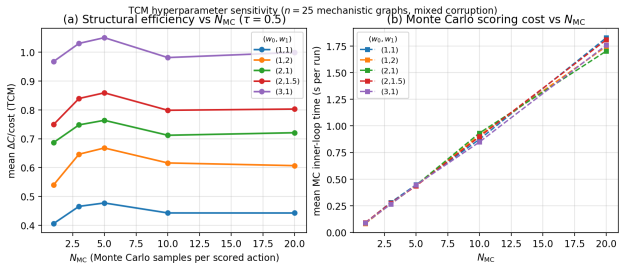


Figure 5. **Sensitivity of mean TCM structural efficiency and Monte Carlo time to N_{MC} at τ=0.5.** Curves distinguish (w_0, w_1) pairs on n=25 stratified mechanistic graphs. Panel (a): mean $\Delta C/\text{cost}$; panel (b): mean total wall time in TCM Monte Carlo inner loops per run.

mechanistic graphs used for the ablation protocol (mixed corruption, budget 8.0, realistic intervention costs, seed 42), we run CycleRank and Entropy once per graph at τ=0.5, then sweep TCM over a fixed grid of $(w_0, w_1) \in \{(1,1), (2,1), (1,2), (2,1.5), (3,1)\}$, $\tau \in \{0.4, 0.5, 0.6\}$, and $N_{\text{MC}} \in \{1, 3, 5, 10, 20\}$, reusing matched RNG seeds per graph.

Figure 5 visualizes the same τ=0.5 slice: larger w_0 with moderate w_1 raises mean structural efficiency, while N_{MC} drives a roughly linear increase in wall-clock time inside Monte Carlo scoring loops (Entropy and CycleRank incur negligible comparable sampling cost). Mean ranks $\overline{\text{rk}}_{\text{eff}}$ and $\overline{\text{rk}}_{\text{F1}}$ average, over graphs, the ordinal rank of TCM among TCM, Entropy, CycleRank, Random on $\Delta C/\text{cost}$ and edge F1 (1 is best).

E. Optional Error-Controlled Discovery Reporting

Purpose and scope. TCM is a cost-aware adaptive *test-selection* policy over dependent edge hypotheses. Its role is to decide which structural hypothesis should be queried next under a finite budget. By itself, TCM is not a multiple-testing correction procedure and does not provide formal error control for the final set of declared edges.

This appendix describes an optional reporting layer that can be placed on top of an acquisition policy. The key separation

is acquisition policy (which hypothesis to test next) versus reporting rule (which hypotheses to declare discovered). TCM is the default acquisition policy considered here, but the same reporting layer can be paired with other acquisition policies. An e-value/eBH procedure can be used for the reporting layer.

The operational object is the declared discovery set

$$D_t \subseteq \mathcal{E}_{\mathrm{eBH}},$$

whose false discovery rate is targeted by the reporting rule under its assumptions, relative to the ground-truth edge set $E^\star$. This is different from controlling global structural correctness of the recovered graph. In particular, eBH can control false discoveries among declared edges under its assumptions, but it does not certify that the full graph $G^\star$ has been recovered under a finite testing budget.

Per-query e-value. For each candidate directed edge (i, j), define the binary structural hypotheses

$$H_{ij}^0 : e_{ij} = 0 \qquad \text{versus} \qquad H_{ij}^1 : e_{ij} = 1.$$

The simulator oracle returns an outcome $o_t \in \{+, -\}$ with sensitivity η and specificity ξ:

$$\mathbb{P}(o_t = + \mid H_{ij}^1) = \eta, \qquad \mathbb{P}(o_t = + \mid H_{ij}^0) = 1 - \xi.$$

Equivalently,

$$\mathbb{P}(o_t = - \mid H_{ij}^1) = 1 - \eta, \qquad \mathbb{P}(o_t = - \mid H_{ij}^0) = \xi.$$

For a queried edge, define the one-step likelihood-ratio statistic

$$e_t = \frac{\mathbb{P}(o_t \mid H_{ij}^1)}{\mathbb{P}(o_t \mid H_{ij}^0)} = \begin{cases} \eta/(1-\xi), & o_t = +, \\ (1-\eta)/\xi, & o_t = -. \end{cases}$$

If the oracle law used in this likelihood ratio matches the simulator that actually generates o_t, then under H_{ij}^0,

$$\mathbb{E}[e_t \mid H_{ij}^0] = \sum_{o \in \{+,-\}} \mathbb{P}(o \mid H_{ij}^0) \frac{\mathbb{P}(o \mid H_{ij}^1)}{\mathbb{P}(o \mid H_{ij}^0)} = 1.$$

Thus, for a single oracle draw, e_t is a valid e-value against H_{ij}^0.

Sequential accumulation. If the same edge is queried repeatedly, we maintain a cumulative statistic

$$E_{ij,t} = \prod_{s \le t:\, (i,j) \text{ queried at } s} e_s,$$

with initialization

$$E_{ij,0} = 1.$$

This cumulative product is a valid e-process under the usual sequential validity conditions. A simple sufficient condition is that, conditional on H_{ij}^0, successive oracle draws for the same edge are conditionally independent of past draws and each factor uses the correct conditional likelihood ratio and, more generally, the product remains valid when the factors form an e-process adapted to the query history.

This assumption matters. If the oracle model is misspecified, meaning the values of η and ξ used in the likelihood ratio differ from the actual oracle error rates, then the expectation under the null need not be bounded by one. Similarly, if repeated oracle outcomes for the same edge are dependent and the product does not account for that dependence, the cumulative statistic may fail to be an e-process. In those cases, the eBH interpretation should be treated as diagnostic rather than guaranteed.

e-BH discovery rule. Let

$$\mathcal{E}_{\mathrm{eBH}} \subseteq \mathcal{E}_{\mathrm{cand}}$$

denote the fixed, pre-declared eBH universe used for reporting discoveries. In the implementation, this universe is defined at initialization and may be a strict subset of all belief keys when an `--efdr-m-max` cap is used. Let

$$m = |\mathcal{E}_{\mathrm{eBH}}|$$

be the size of this eBH universe. At time t, let

$$\{E_{ij,t} : (i,j) \in \mathcal{E}_{\mathrm{eBH}}\}$$

denote the current cumulative e-values for edges in the reporting universe. Let

$$E_{(1),t} \ge E_{(2),t} \ge \cdots \ge E_{(m),t}$$

be these values sorted in decreasing order. At target level $q \in (0, 1)$, eBH selects

$$\widehat{k}_t = \max\left\{k \in \{1, \ldots, m\} : E_{(k),t} \ge \frac{m}{qk}\right\}.$$

The implementation returns the first $\widehat{k}_t$ edges in this sorted order,

$$D_t^{\mathrm{eBH}} = \{(i,j)_{(1)}, \ldots, (i,j)_{(\widehat{k}_t)}\},$$

with

$$D_t^{\mathrm{eBH}} = \emptyset$$

if no such $\widehat{k}_t$ exists. This matches the sorted-list implementation of eBH with the fixed reporting universe size m. If several edges tie exactly at the cutoff, this may differ slightly from the tie-inclusive set

$$\{(i,j) : E_{ij,t} \ge E_{(\widehat{k}_t),t}\}.$$

This is the eBH procedure for multiple testing with e-values (81; 82).

Algorithm 2 Acquisition proposes tests; eBH reports discoveries

Initialize beliefs over candidate edges $\mathcal{E}_{\text{cand}}$
Pre-declare an eBH reporting universe $\mathcal{E}_{\text{eBH}} \subseteq \mathcal{E}_{\text{cand}}$
Set $m \leftarrow |\mathcal{E}_{\text{eBH}}|$ and $E_{ij} \leftarrow 1$ for all $(i,j) \in \mathcal{E}_{\text{eBH}}$
Fix target FDR level $q \in (0,1)$
while budget remains and feasible actions exist **do**
  Build the current belief graph using the implementation threshold $p_{ij} > 0.5$
  Compute topological complexity using the topology threshold used by `compute_complexity`
  Select action a_t using the acquisition policy ▷ TCM is the default instantiation
  for each queried edge $(i,j) \in a_t$ **do**
    Draw oracle outcome $o_t \in \{+,-\}$
    Update posterior belief p_{ij} by Bayes' rule
    if $(i,j) \in \mathcal{E}_{\text{eBH}}$ **then**
      Compute $e_t = \mathbb{P}(o_t \mid H^1_{ij})/\mathbb{P}(o_t \mid H^0_{ij})$
      Update cumulative e-value $E_{ij} \leftarrow E_{ij} \cdot e_t$
    end if
  end for
  $D_t^{\text{naive}} \leftarrow \{(i,j) : p_{ij} > 0.5\}$ ▷ diagnostic baseline, not FDR-valid
  $D_t^{\text{eBH}} \leftarrow \text{EBH}(\{E_{ij}\}_{(i,j)\in\mathcal{E}_{\text{eBH}}}, m, q)$
end while

Acquisition selection versus eBH discovery. In the default TCM instantiation, tests are selected by maximizing a Monte Carlo estimate of expected topological simplification per unit cost:

$$a_t^\star = \arg\max_{a\in\mathcal{A}_t} \frac{\widehat{\mathbb{E}}[C(K_t) - C(K_{t+1}) \mid a]}{\text{Cost}(a)}.$$

In the implementation, this expectation is approximated with N Monte Carlo samples rather than computed analytically. The eBH layer does not feed back into this acquisition score unless explicitly coupled to the policy. It updates cumulative e-values only for queried edges in $\mathcal{E}_{\text{eBH}}$, and applies eBH with the same fixed m to decide which edges are declared discovered.

Naive thresholding and topology thresholds. The naive discovery baseline in the implementation uses the hard threshold

$$D_t^{\text{naive}} = \{(i,j) : p_{ij} > 0.5\}.$$

This is a diagnostic posterior-thresholding rule, not an FDR-valid multiple-testing procedure under adaptive querying. The topology threshold used in `compute_complexity` also defaults to 0.5, but it is a separate parameter. Therefore, if the topology threshold is changed away from 0.5, the naive discovery set and the graph used by the intervention generator need not coincide with the topology threshold used for complexity computation.

Empirical FDP and recall reporting. For diagnostic evaluation in simulation, we compute the realized false discovery proportion of any declared set D_t:

$$\text{FDP}(D_t) = \frac{|\{(i,j) \in D_t : (i,j) \notin E^\star\}|}{\max(|D_t|, 1)}.$$

We also compute recall of true edges inside the reported discovery set:

$$\text{Recall}(D_t) = \frac{|D_t \cap E^\star|}{|E^\star|}.$$

If $D_t = \emptyset$, the FDP is defined as 0 by the denominator convention above, and recall is also 0.

When aggregating across simulator instances, curves report the mean realized FDP or mean recall across instances at each spent budget checkpoint. For a budget value b, we read off, for each mechanism instance, the last recorded discovery set at cost at most b, then average the corresponding FDP or recall values across instances. Shaded bands show a normal-approximation confidence band,

$$\text{mean} \pm 1.96 \times \text{SE}.$$

These finite-sample mean-FDP curves estimate

$$\text{FDR} = \mathbb{E}[\text{FDP}]$$

under independent replications of the full simulator, but they do not by themselves prove finite-sample FDR control. The dashed line in the plots marks the target eBH level q used as a reference.

Informal guarantee. Fix $q \in (0,1)$ and a fixed eBH reporting universe $\mathcal{E}_{\text{eBH}}$. Suppose that for every absent edge $(i,j) \notin E^\star$ in $\mathcal{E}_{\text{eBH}}$, the cumulative statistic $E_{ij,t}$ used by eBH is a valid e-process under the actual sequential oracle model. Then applying eBH at level q to the current collection of edge-level e-values controls the false discovery rate of the declared discovery set under the e-value model assumptions (82). This guarantee concerns false discoveries among declared edges; it does not imply that the entire graph has been recovered, nor that TCM's topological acquisition rule is itself an error-control procedure.

Acquisition versus reporting in the mechanistic pool. Figure 6 summarizes a controlled separation experiment on the same corruption–oracle simulator used elsewhere in the paper. For each ground-truth mechanism graph, we reset the simulator RNG before corruption so that the initial belief graph is identical across policies, instantiate a fixed eBH reporting universe with identical hyperparameters across runs, and rerun the adaptive testing episode while swapping only the acquisition policy (TCM, ENTROPY, CYCLERANK,

RANDOM), with the reporting layer otherwise held constant. Cumulative likelihood-ratio e-values are updated only for edges in the pre-declared eBH universe, and discoveries are formed by applying eBH at level q to that universe with the same fixed m at each budget checkpoint. The naive baseline instead thresholds all edge posteriors at 0.5.

Across policies, the eBH mean FDP curves remain near or below the plotted reference level q, whereas naive mean FDP is substantially larger and varies with the query path induced by each acquisition rule, especially at low spent budget. Recall shows the complementary trade-off: naive discovery sets cover many true edges and therefore achieve high mean recall, while eBH mean recall is modest and policy-dependent, reflecting the intentional conservatism of eBH at moderate budgets when many hypotheses are monitored. Overall, the experiment supports the methodological point that changing acquisition alone changes realized error and power profiles, even when the reporting rule is fixed.

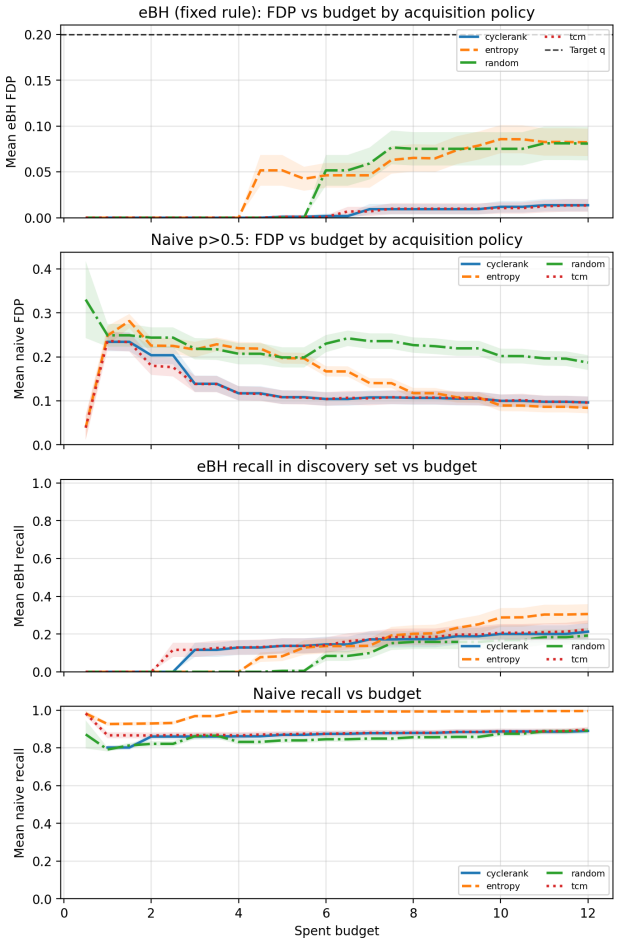
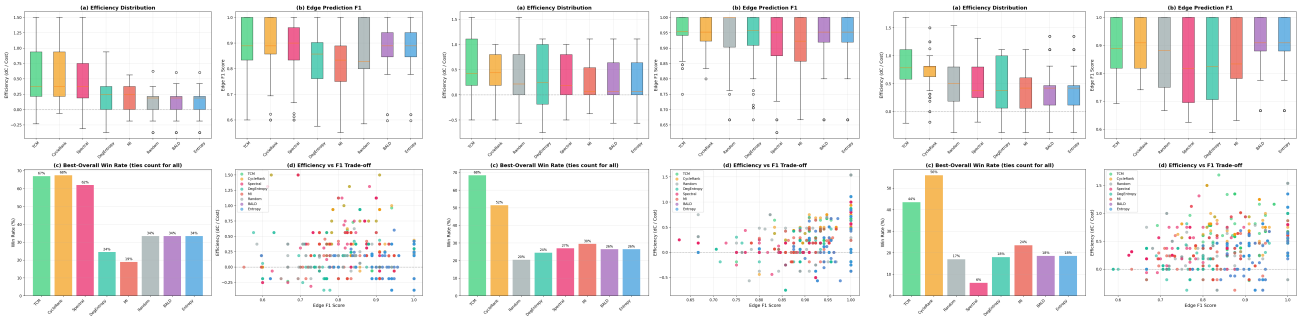
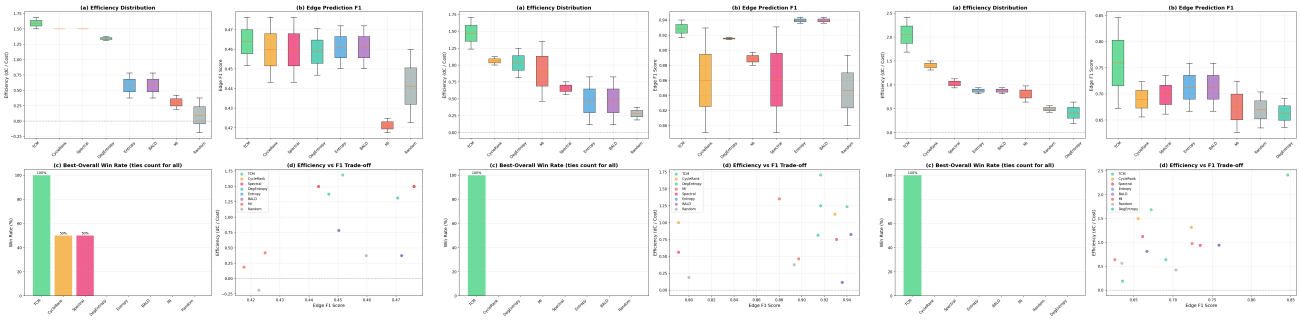


Figure 6. **Separating acquisition from reporting under a fixed eBH rule.** Mean realized FDP and recall over spent budget for four acquisition policies (TCM, ENTROPY, CYCLERANK, RANDOM) under a shared eBH reporting layer and a naive posterior-thresholding diagnostic. Shaded bands show $\pm 1.96\times$SE across stratified mechanistic instances. The eBH curves use a fixed pre-declared reporting universe and target level q; the naive rule declares all candidate edges with $p_{ij} > 0.5$. The plot illustrates how acquisition affects realized error and power profiles under a fixed reporting rule; it is not a proof of FDR control.



(a) Mechanistic, cycle-dominated. **(b)** Mechanistic, fragmentation-dominated. **(c)** Mechanistic, mixed.

Figure 7. **Mechanistic regime breakdown (`mech-USPTO-31k`).** Each panel summarizes one corruption regime with the same four views as the overall figure: efficiency ($\Delta C/\text{Cost}$) distributions, edge F1, best-overall win rate (ties credited to all tied methods), and an efficiency–F1 scatter. These plots complement the mean summary in Table 1 by showing dispersion and head-to-head frequency at the per-mechanism level.



(a) Causal pool, cycle. **(b)** Causal pool, fragment. **(c)** Causal pool, mixed.

Figure 8. **Causal pool regime breakdown (CHILD + INSURANCE).** Regime-separated summaries for the causal multi-graph protocol (same plotting template as Figure 7). Means over the two DAGs appear in Table 1; these panels show per-instance dispersion consistent with that aggregate.