OPTIMAL CORRECTION SETS FOR PROVABLE REDRESS OF CONTESTABLE CAUSAL DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Constraint-based causal discovery relies on conditional-independence (CI) tests that may conflict in finite samples, leaving no causal graph that satisfies them all. Causal Assumption-based Argumentation (ABA) makes these tests explicit as defeasible commitments, and offers correspondence guarantees with a subset of them, but its existing repair procedure releases facts heuristically and offers no formal response to a user's contestation of a release. We argue that contestable causal discovery requires an inspectable repair object, a justification for it, and a predictable revision procedure for such contestations. We therefore formulate CI-fact repair as a minimum-cost correction problem and implement it in Answer Set Programming (ASP). Building on Causal ABA, we provide a compact and efficient ASP encoding of causal consistency, with soundness and completeness guarantees, and a two-way projection correspondence between optimal stable models and minimum-cost correction sets. These results establish fidelity to retained facts, subset-minimality and cost optimality of the repair, conflict-based justifications for released facts, and formally guaranteed redress under user contestations. An empirical evaluation supports the formal account: OptABA-PC retains more correct CI statements, reduces ambiguity in the compatible output causal graphs, while keeping competitive reconstruction performance on common benchmarks with limited number of nodes.

1 INTRODUCTION

Causal discovery supports scientific understanding and decision making by turning observational and/or interventional data into hypotheses about the underlying causal mechanisms. It is a central ingredient in tasks such as explanation, prediction under interventions, and policy evaluation (Pearl, 2009). A wide range of approaches exists, including constraint-based methods relying on independence testing, score-based methods that optimise a goodness-of-fit criterion over graphs, and functional causal model approaches that leverage assumptions on the data-generating mechanism; see e.g. (Glymour et al., 2019; Zanga et al., 2022).

Constraint-based algorithms such as PC (Spirtes et al., 2000) infer structure from conditional-independence (CI) information. Causal ABA (Russo et al., 2024) instead treats selected CI facts as defeasible commitments that every returned graph must satisfy if enforced, exposing data-derived premises to causal discovery claims and a point of contestation. In this context, a *contestation* asks the system to reconsider a result by revising those input facts. Computational argumentation supports this view: contestation should lead beyond static explanation to reliable revision when redress is warranted (Leofante et al., 2024); argumentative decision making captures that reliability through contestability guarantees under user-driven changes (Freedman et al., 2025).

Finite-sample CI tests can yield mutually incompatible outcomes, leaving no DAG that satisfies them Russo et al. (2024); Faltenbacher et al. (2026). Majority-PC (MPC) (Colombo & Maathuis, 2014), an improved PC variant for finite samples, is sound under standard assumptions with an oracle for error-free CI judgements, but its finite-sample output need not satisfy all recorded test outcomes. Causal ABA instead restores coherence by releasing facts in strength order until the remaining commitments are consistent. This procedure guarantees neither a minimum-cost repair nor a predictable response when a user contests a released fact or its cost. The problem is therefore to identify a justified repair of conflicting CI evidence and ensure predictable revision under contestations.

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.

Example 1 (Causal discovery)**.** *Consider a simple socioeconomic setting with Education (E), Race (R), Occupation (O) and Income (I) with ground-truth directed acyclic graph (DAG) in Fig. 1. When fed data generated from this DAG, MPC runs* 14 *CI tests (see Ex. 3) and uses them sequentially to remove and orient edges. With finite-sample noise, it may report* mutually incompatible *outcomes. For instance, it includes the high-strength independences $E \perp\!\!\!\perp O \mid \{I\}$ and $O \perp\!\!\!\perp I$, even though it also reports dependencies such as $E \not\perp\!\!\!\perp O$ and $E \not\perp\!\!\!\perp I$. These four statements already cannot be simultaneously satisfied by any DAG: if E and O are dependent but become independent after conditioning on I, then I must block every active E–O path, which implies $O \not\perp\!\!\!\perp I$—contradicting $O \perp\!\!\!\perp I$. Nevertheless, MPC returns a non-empty graph (the single undirected edge $O - R$), yet neither identifies the incompatible constraints nor records which it effectively discards.*

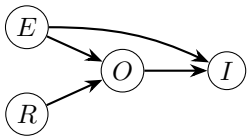

Figure 1: Example DAG.

We address this problem by making the correction set the *redress* object and proving optimal repair. We build on explanation and repair in logic programming (Reiter, 1987; Liffiton & Sakallah, 2008; Alviano et al., 2023), particularly minimal unsatisfiable subsets (MUSes) and minimal correction subsets (MCSes). A MUS isolates an irreducible conflict; an MCS is a minimal set of facts whose release restores coherence. In Causal ABA, these objects explain conflicting CI outcomes and input causal relations, and identify the releases available as redress. We preserve PC's soundness and completeness guarantees, Causal ABA's CI–DAG correspondence guarantees and add minimum-cost repair, conflict certificates, and guaranteed hard and graded redress. A user can contest a release by requiring the fact's retention or increasing its release cost and re-solving. Concretely, we: (i) formulate Causal ABA's released CI facts as an explicit repair; (ii) prove a two-way correspondence between optimal MCSes and optimal stable models that allows efficient computation with state-of-the-art solvers; (iii) contribute a compact and efficient Bayes-ball encoding (Shachter, 1998); (iv) establish conflict certificates and guaranteed hard and graded contestability; (v) implement the optimisation as OptABA-PC; and (vi) compare it with Causal ABA and standard baselines, generally improving fact selection and/or graph reconstruction while providing redress guarantees.

2 Related Work

The relationship between inconsistency, repairs, and hitting sets is well established. In model-based diagnosis, diagnoses are hitting sets of conflicts (Reiter, 1987); in constraint systems and SAT, MUSes expose irreducible conflicts and MCSes repair them by hitting those conflicts (Liffiton & Sakallah, 2008). Related ideas appear in optimisation as maximum feasible subsystem and IIS-cover problems, where one deletes a minimum-weight set of constraints to restore feasibility (Chinneck & Dravnieks, 1991; Amaldi et al., 2003), and in implicit hitting-set methods, which alternate between a feasibility oracle returning an unhit conflict and an optimiser selecting a new hitting set (Moreno-Centeno & Karp, 2013). Core-guided MaxSAT solvers operationalise the same pattern over soft clauses (Morgado et al., 2013). Our use of MUS/MCS terminology therefore does not introduce a new optimisation paradigm; rather, it makes explicit what the repair object is in Causal ABA.

In causal discovery, logic-based methods have used SAT to enforce constraints from overlapping observational or interventional data and query graph features common to all solutions (Triantafilou et al., 2010; Triantafillou & Tsamardinos, 2015). ASP-based approaches infer causal features from noisy evidence through weighted optimisation over ancestral relations (Magliacane et al., 2016) or greedy selection of constraints consistent with semi-Markov causal models (Borboudakis & Tsamardinos, 2016). These methods target causal conclusions or graph-level agreement; they do not make the correction set of released facts the explicit output.

SAT/ASP has also been used to encode d-separation exactly. Hyttinen et al. (2013) introduce variables for directed and bidirected edges and bounded d-connecting walks, then compute backbone features shared by all satisfying cyclic/latent causal graphs; inconsistent inputs yield no repair. Hyttinen et al. (2014) handle noisy inputs by deriving `fail` atoms when a graph violates a CI constraint and minimising their total weight. They study several weighting schemes; we call the DAG-restricted empirical implementation of this objective ASPCR-DAG. Our proposed OptABA-PC likewise uses weighted optimisation, but its CI-strength weights are release costs, its released facts form an explicit correction set, and every retained fact is satisfied by every returned graph. The encoding of Hyttinen et al. (2014) instead permits penalised `fail` atoms in an optimal model. Compared to ASPCR, Causal ABA targets DAGs, which permits the compact Bayes-ball encoding while preserving an argumentative reading of retained facts.

Hyttinen et al. (2017) recast the same weighted graph-disagreement objective as MaxSAT: weighted unit soft clauses represent CI constraints, hard clauses connect them to graph satisfaction, a SAT solver extracts unsatisfiable cores, and an integer program computes a minimum-cost hitting set of the accumulated cores. This is the closest prior optimisation architecture to OptABA-PC. Its cores and hitting sets already provide conflict/repair machinery; our additional results connect explicit released-fact sets to optimal stable models, allowing computation with ASP solvers, and establish fact enforcement and contestability guarantees. Rantanen et al. (2020) use branch-and-bound with partial-graph propagation, tailored branching, and core-derived bounds.[1] Complementary SAT/ASP work weakens assumptions or extends the graph semantics (Zhalama et al., 2017; 2019).

In Causal ABA (Russo et al., 2024), accepted CI facts are not merely soft clauses or post-hoc `fail` atoms: they are enforceable ABA/ASP facts whose satisfaction is guaranteed in every returned stable model, yielding traceability from data-derived facts to outrput causal graphs. OptABA-PC makes the released facts first-class: by placing weak constraints on unselected objective atoms, optimal stable models correspond to optimal correction sets of the Causal ABA program (implemented in ASP). Whereas existing ASP work on MUSes studies subset-minimality, enumeration, and cautious reasoning (Alviano et al., 2023), we connect optimal stable models to minimum-cost correction sets and leverage these to establish contestability guarantees of our causal discovery method.

3 PRELIMINARIES

We consider a finite set of variables $\mathbf{V}$. We write $X \perp\!\!\!\perp Y \mid \mathbf{Z}$ for conditional independence between variables X and Y given conditioning set $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}$, and $X \not\perp\!\!\!\perp Y \mid \mathbf{Z}$ for dependence ($X \perp\!\!\!\perp Y$ and $X \not\perp\!\!\!\perp Y$ if $\mathbf{Z} = \emptyset$). In our causal-discovery setting, a CI fact is a ground assertion of one such relation, obtained from a statistical CI test for a significance threshold α.

A causal graph is a DAG $G = (\mathbf{V}, E)$ whose edges represent direct causal relations. A node Z on a path is a *collider* on that path if the two adjacent path edges both have arrowheads into Z, as in $X \to Z \leftarrow Y$; otherwise it is a *non-collider*. A node Y is a descendant of X if there is a directed path from X to Y. A collider is *open* given a conditioning set $\mathbf{Z}$ when the collider itself or at least one of its descendants belongs to $\mathbf{Z}$; otherwise it is closed. DAGs entail CI relations through d-separation (Pearl, 1989): a path between X and Y is active given $\mathbf{Z}$ if every non-collider on the path is outside $\mathbf{Z}$ and every collider is open given $\mathbf{Z}$; X and Y are d-connected given $\mathbf{Z}$ if some such active path exists, and d-separated otherwise. Observational data identify at most a Markov equivalence class (MEC) (Richardson & Spirtes, 2002), represented by a *completed partially* DAG (CPDAG) (Chickering, 2002a) whose directed edges are shared by all DAGs in the class and whose undirected edges are reversible.

3.1 ANSWER SET PROGRAMMING

Answer Set Programming (ASP) represents a problem by rules whose stable models are its solutions. A ground normal rule has form h `:-` $l_1, \ldots, l_n$, where each body literal is an atom a or default negation `not` a, which holds when a is absent; a headless rule forbids its body. We write $AS(P)$ for the stable models of P and call P *coherent* when $AS(P) \neq \emptyset$. We use choice rules $\{a\}$ to allow selecting a and weak constraints `:~` $B.\ [w@l, o]$ to charge cost w when B holds (Calimeri et al., 2020). The identifier o in the weak-constraint tuple prevents equal-weight facts from collapsing into one occurrence. With priority l set to 1 for all constraints, an optimal stable model simply minimises the total cost. App. A gives more background on ASP and its reduct-based stable model semantics (Gelfond & Lifschitz, 1988).

3.2 ABA AND CAUSAL ABA

An assumption-based argumentation framework (ABAF) is a tuple $(\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\cdot})$, where $\mathcal{L}$ is a language, $\mathcal{R}$ is a set of rules over $\mathcal{L}$, $\mathcal{A} \subseteq \mathcal{L}$ is a set of assumptions, and $\bar{\cdot}$ maps each assumption $a \in \mathcal{A}$ to its contrary $\overline{a} \in \mathcal{L}$ (Bondarenko et al., 1997; Cyras et al., 2017). A set $S \subseteq \mathcal{A}$ derives q, written $S \vdash q$, if q follows from S by the rules. It attacks an assumption a when $S' \vdash \overline{a}$ for some $S' \subseteq S$. A stable *extension* is a conflict-free set of assumptions that attacks every assumption outside it. This formalism is useful for causal discovery because assumptions are defeasible commitments: they may be accepted or rejected based on their interrelations.

Causal ABA (Russo et al., 2024) uses assumptions for candidate graph choices (directed edges) and CI facts, and rules encoding acyclicity and d-separation. A stable extension of Causal ABA

[1] No public implementation was identified for these latter methods, so we compare to (Hyttinen et al., 2014).

corresponds to a candidate DAG together with the CI facts it entails via d-separation. Enforcing a CI fact means requiring it to be in the accepted set and rejecting any stable extension whose graph violates it; releasing a fact means allowing it to be absent so coherence can be restored. Under the standard correspondence between ABA under stable semantics and ASP stable models (Rapberger et al., 2024), this ABAF can be solved as an ASP program. Hence Causal ABA inherits both the argumentation reading of assumptions and the computational machinery of ASP. Causal ABA may admit multiple stable extensions for the same enforced CI facts, with corresponding DAGs spanning one or more MECs. This multiplicity is informative: the CI–DAG correspondence guarantees that every candidate satisfies those facts, while differing graphs expose the structural ambiguity they leave unresolved. These sets describe uncertainty conditional on the enforced facts, without implying calibrated confidence or guaranteed coverage of the generating MEC.

Example 1 highlights the situation Causal ABA is designed to expose: the empirical CI facts may be mutually incompatible, yet we still want returned graphs to satisfy a subset of the facts that remain enforced and to record the ones released.

Example 2 (Causal ABA)**.** *Continuing Ex. 1, Causal ABA's grounded ASP encoding would include:*

```
collider(o,e,r)  :- arrow(e,o), arrow(r,o), e!=r.
nb(o,e,r,Z)      :- in(o,Z), collider(o,e,r).
ap(e,r,p_eor,Z)  :- arrow(e,o), arrow(r,o), nb(o,e,r,Z).
ap(e,r,p_eior,Z) :- arrow(e,i), arrow(i,o), arrow(r,o), nb(o,e,r,Z).
dep(e,r,Z)       :- ap(e,r,_,Z).
indep(e,r,Z)     :- not ap(e,r,p_eor,Z),not ap(e,r,p_eior,Z).
```

Intuitively, `arrow(x,y)` *encodes a directed edge* $X \to Y$*;* `in(v,Z)` *encodes membership in the conditioning set;* `collider` *and* `nb` *characterise whether a node blocks a path given* $\mathbf{Z}$*; and* `ap(x,y,p,Z)` *witnesses an active path, where labels such as* `p_eor` *and* `p_eior` *are mnemonic identifiers for the corresponding path shapes. The atoms* `dep`/`indep` *are then derived from* `ap`*. Empirical CI tests are provided as input facts (e.g.,* `dep(o,r,{})` *for* $O \not\perp\!\!\!\perp R$ *and* `indep(r,i,{o})` *for* $R \perp\!\!\!\perp I \mid \{O\}$*) and enforced by rejecting any stable model whose derived* `dep`/`indep` *atoms disagree with the input. For* $\mathbf{Z} = \{O\}$*, conditioning on the collider* O *opens the path* $E \to O \leftarrow R$*, so the encoding derives* `dep(e,r,{o})`*, representing* $E \not\perp\!\!\!\perp R \mid O$*.*

If the enforced facts are jointly inconsistent, the ASP program becomes incoherent (no stable model) and a repair step must decide which facts to release.

3.3 CAUSAL ABA-PC

Causal ABA-PC (Russo et al., 2024) is a concrete instantiation of Causal ABA in which the defeasible constraints are CI facts obtained from MPC (Colombo & Maathuis, 2014). Causal ABA-PC assigns each fact a *strength* derived from its test p-value (Casella & Berger, 2002) and then incrementally releases low-strength facts until coherence is restored: a simple heuristic baseline repair strategy.

Example 3 (Causal ABA-PC)**.** *Continuing Ex. 1, consider a Causal ABA-PC run on variables* (E, R, O, I) *whose MPC CI tests produce a trace of* `dep`/`indep` *facts, sorted by descending strength* S *and labelled as true (T) or false (F) w.r.t. the ground-truth DAG in Fig. 1, as follows:*

```
indep(e,o,{i}) S=1.00 (F)        dep(e,i,{r})   S=0.56 (T)
indep(r,i,{o}) S=1.00 (F)        dep(r,i,{})    S=0.54 (T)
dep(o,r,{})    S=0.99 (T)        indep(e,i,{o}) S=0.53 (F)
dep(o,r,{e})   S=0.95 (T)        indep(r,i,{e}) S=0.53 (F)
dep(e,o,{})    S=0.92 (T)        dep(o,r,{i})   S=0.52 (T)
dep(e,o,{r})   S=0.92 (T)        indep(o,i,{})  S=0.51 (F)
dep(e,i,{})    S=0.84 (T)        indep(e,r,{})  S=0.50 (T)
```

MPC uses all 14 *tests (including* 5 *wrong ones), producing a CPDAG with a single undirected edge* $(O - R)$*. ABA-PC releases facts in reverse strength order until the nine entries before the dashed separator remain enforced (*7 *true and* 2 *false), yielding one compatible DAG with one correct edge* $(O \to I)$ *and two wrong ones* $(O \to R,\ I \to E)$*.*

Example 3 illustrates that enforcing all CI facts may be impossible, and that the heuristic release strategy of Causal ABA-PC may discard true facts while retaining false ones. We now show how MUSes and MCSes can be used to identify sets of facts that carry desirable properties for repair.

3.4 MUSES AND MCSES

Explaining and repairing incoherence is a classic theme in diagnosis of ASP programs (Reiter, 1987; Liffiton & Sakallah, 2008), and it naturally applies here because noisy CI-test outcomes need not be jointly satisfiable.

MUSes and MCSes are defined relative to a base ASP program P and a distinguished set O of *objective atoms*. For $\mathrm{S} \subseteq \mathrm{O}$, let $\mathsf{enforce}(\mathrm{P}, \mathrm{S}, \mathrm{O})$ denote the program obtained from P by (i) adding a choice rule over atoms in O and (ii) forcing every atom in S to be true. Concretely:

$$\mathsf{enforce}(\mathrm{P}, \mathrm{S}, \mathrm{O}) := \mathrm{P} \cup \{\{o\}. \mid o \in \mathrm{O}\} \cup \{\texttt{:- not } o \mid o \in \mathrm{S}\}.$$

Definition 1 (Alviano et al. 2023)**.** *Let* P *be a program and* O *be a set of objective atoms. Then* $\mathrm{S} \subseteq \mathrm{O}$ *is an* unsatisfiable set *if* $\mathsf{enforce}(\mathrm{P}, \mathrm{S}, \mathrm{O})$ *is incoherent.* S *is a minimal unsatisfiable set (MUS) if no strict subset* $\mathrm{S}' \subset \mathrm{S}$ *is an unsatisfiable set.*

Definition 2 (Alviano et al. 2023)**.** *Let* P *be a program and* O *be a set of objective atoms. Then* $\mathrm{S} \subseteq \mathrm{O}$ *is a* correction set *if* $\mathsf{enforce}(\mathrm{P}, \mathrm{O} \setminus \mathrm{S}, \mathrm{O})$ *is coherent.* S *is a minimal correction set (MCS) if no strict subset* $\mathrm{S}' \subset \mathrm{S}$ *is a correction set.*

MUSes pinpoint irreducible conflicts, whereas MCSes identify subset-minimal repairs. Write $MUS(\mathrm{P}, \mathrm{O})$ and $MCS(\mathrm{P}, \mathrm{O})$ for the families of all MUSes and MCSes of P w.r.t. O, respectively. Every MCS must hit every MUS; indeed, $MCS(\mathrm{P}, \mathrm{O})$ is exactly the family of minimal hitting sets of $MUS(\mathrm{P}, \mathrm{O})$, and conversely (Alviano et al., 2023, Thm. 5.3). Their number may be exponential.

Example 4 (MUS/MCS intuition)**.** *Consider a program* P*:*

```
a :- not b, o(1).  b :- not c, o(2).  c :- not a, o(3).  :- not d, o(4).
```

with objective $\mathrm{O} = \{\texttt{o(1)}, \texttt{o(2)}, \texttt{o(3)}, \texttt{o(4)}\}$*. Enforcing all of* O *makes* $\mathsf{enforce}(\mathrm{P}, \mathrm{O}, \mathrm{O})$ *incoherent, but for* two independent reasons*: enforcing* `o(4)` *forces* `d` *even though it cannot be derived, and enforcing* `o(1)`, `o(2)`, `o(3)` *activates the negative loop over* `a`, `b`, `c`*. Hence the MUSes are* $\{\texttt{o(4)}\}$ *and* $\{\texttt{o(1)}, \texttt{o(2)}, \texttt{o(3)}\}$*, while the MCSes are the minimal hitting sets* $\{\texttt{o(1)}, \texttt{o(4)}\}$*,* $\{\texttt{o(2)}, \texttt{o(4)}\}$*, and* $\{\texttt{o(3)}, \texttt{o(4)}\}$*.*

4 OPTIMAL CORRECTION SETS IN ASP

Let $CS(\mathrm{P}, \mathrm{O})$ contain all correction sets for P w.r.t. O. We assign each objective atom a strictly positive release cost, $\mathrm{W} = \{(o, w_o) \mid o \in \mathrm{O},\ w_o > 0\}$, and write $\mathcal{C}(\mathrm{R}, \mathrm{W}) = \sum_{o \in \mathrm{R}} w_o$. We seek a minimum-cost coherent repair without enumerating exponentially many MCSes.

Definition 3 (Optimal MCS)**.** *Let* P *be a program,* O *a set of objective atoms, and* W *weights over* O*. An MCS* $\mathrm{S} \in MCS(\mathrm{P}, \mathrm{O})$ *is* optimal *(w.r.t.* W*) if for every* $\mathrm{S}' \in MCS(\mathrm{P}, \mathrm{O})$ *it holds that* $\mathcal{C}(\mathrm{S}, \mathrm{W}) \leq \mathcal{C}(\mathrm{S}', \mathrm{W})$.

Lemma 1 (Positive-cost minimality)**.** *Let* P *be a finite ground program, let* O *be finite, and let* $\mathrm{W} = \{(o, w_o) \mid o \in \mathrm{O},\ w_o > 0\}$*. Every minimum-cost member of* $CS(\mathrm{P}, \mathrm{O})$ *is an optimal MCS with respect to* W*.*

All proofs are in Appendix C–F.

Subset-minimal solutions can be enumerated using WASP (Alviano et al., 2023). We instead expose each objective atom as a choice and minimise release costs via weak constraints, using *clingo* (Gebser et al., 2019) in our implementation. For a program P, objective atoms O, and weights W, define

$$\mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W}) = \mathrm{P} \cup \{\{o\}. \mid o \in \mathrm{O}\} \cup \{\texttt{:\sim not } o.\ [w_o@1, o] \mid (o, w_o) \in \mathrm{W}\}.$$

For a stable model M, let its released-fact projection be $\rho(M) = \mathrm{O} \setminus M$.Consequently, the weak-constraint cost of M is exactly $\mathcal{C}(\rho(M), \mathrm{W})$.

Example 5 (Optimal MCS)**.** *In Ex. 4, assigning costs* $5, 10, 15, 20$ *to* `o(1)`, …, `o(4)` *gives the three repairs costs* $25, 30, 35$*; releasing* `o(1)` *and* `o(4)` *is therefore uniquely optimal. For the same instance, an optimal model is* $M = \{\texttt{o(2)}, \texttt{o(3)}, \texttt{c}\}$*, so* $\rho(M) = \{\texttt{o(1)}, \texttt{o(4)}\}$*; the fully instantiated program is in Appendix D.*

Theorem 1 (Projection correspondence)**.** *Let* P *be a finite ground program,* O *a finite set of objective atoms, and* W *assign strictly positive release weights to all atoms in* O*. Then:*

$(\Rightarrow)$ *If* M *is an optimal stable model of* $\mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W})$*, then* $\rho(M)$ *is an optimal MCS (Def. 3), and the weak-constraint cost of* M *is* $\mathcal{C}(\rho(M), \mathrm{W})$*.*

$(\Leftarrow)$ *For every optimal MCS* R*, there exists an optimal stable model* M *of* $\mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W})$ *with* $\rho(M) = R$*.*

Distinct optimal stable models may have the same released-fact projection.

Proposition 1 (Conflict justification)**.** *Let* P *be a finite ground program and* O *a finite set of objective atoms. For every* $\mathrm{R} \in MCS(\mathrm{P}, \mathrm{O})$ *and every* $f \in \mathrm{R}$*, there is a MUS* $U_f \in MUS(\mathrm{P}, \mathrm{O})$ *such that* $U_f \cap \mathrm{R} = \{f\}$*.*

Thus every released objective atom has a conflict certificate containing no other released fact. Reinstating that atom while retaining the others in the certificate restores incoherence. Theorem 1 makes minimum-cost repair a standard ASP optimisation task, enabling the use of optimised solvers without first enumerating all MCSes.The formulation makes repair an inspectable decision that can be revised through fact enforcement or release costs while reusing the underlying consistency encoding. We next instantiate this construction for Causal ABA, where the commitments are CI facts.

5 OPTABA-PC

5.1 BAYES-BALL ENCODING OF CAUSAL ABA

We separate the inherited Causal ABA machinery from our replacement for its active-path computation. For a fixed variable set $\mathbf{V}$ and finite set O of valid CI objective atoms, let $\mathcal{S}_{\mathrm{O}}$ contain the conditioning-set identifiers occurring in O. The inherited $\mathrm{P}_{\mathrm{core}}^{\mathrm{CABA}}$ supplies the variable and conditioning-set vocabulary, CI symmetry and consistency, candidate-DAG generation, adjacency, directed reachability, and acyclicity; $\mathrm{P}_{\mathrm{BB}}^{\mathrm{new}}$ computes d-connection and enforces selected CI facts. Their union is the finite ground base program

$$\mathrm{P}_{\mathrm{CA}}(\mathbf{V}, \mathrm{O}) = \mathrm{P}_{\mathrm{core}}^{\mathrm{CABA}}(\mathbf{V}, \mathcal{S}_{\mathrm{O}}) \cup \mathrm{P}_{\mathrm{BB}}^{\mathrm{new}}(\mathbf{V}, \mathrm{O}). \tag{1}$$

For a fixed instance write P_{CA}. The core chooses $X \to Y$, $Y \to X$, or no edge for each unordered pair, yielding one candidate DAG per stable model; its rules are in Listings 1–3 in Appendix B. Objective choices and weak constraints are added by enforce and opt.

The original Causal ABA encoding derives `dep/indep` by materialising active paths for each queried triple $(X, Y, \mathbf{Z})$. This is faithful to d-separation but expensive, since the number of simple paths can be exponential. We therefore replace the path-enumerating modules with an ASP implementation of the Bayes-ball algorithm (Shachter, 1998). Bayes-ball does not enumerate paths; for a fixed query $(X, Y, \mathbf{Z})$, it propagates a marker through the current DAG according only to the current node, the direction from which the marker arrived, and whether the node is in, or has a descendant in, $\mathbf{Z}$. Bayes-ball is instantiated for input CI queries and their symmetric counterparts. Its two states record arrival from a child or parent; reaching the target derives `ap3(X,Y,Z)`, a d-connection witness. The final constraints require this witness for an enforced dependence and forbid it for an enforced independence. Listing 4 in Appendix B.2 gives the complete encoding.

Lemma 2 (Bayes-ball encoding correctness)**.** *Let* $\mathbf{V}$ *be finite, let* $G \in \mathrm{DAG}(\mathbf{V})$ *fix the* `arrow` *atoms, and let* `qpair(X,Y,Z)` *encode a valid query with distinct endpoints* $X, Y \in \mathbf{V} \setminus \mathbf{Z}$*. In the least fixed point of the grounded propagation rules in Listing 4,* `ap3(X,Y,Z)` *is derived iff* X *and* Y *are* d*-connected given* $\mathbf{Z}$ *in* G*.*

For a stable model M of a Causal ABA instance using P_{CA}, define its *graph projection* by

$$\pi_G(M) = (\mathbf{V}, E_M), \qquad E_M = \{(X, Y) \in \mathbf{V}^2 \mid \mathtt{arrow}(X, Y) \in M\}.$$

This discards the objective, reachability, and Bayes-ball atoms and differs from the released-fact projection $\rho(M)$.

Corollary 1 (Graph fidelity)**.** *For the finite sets* $\mathbf{V}$ *and* O *defining* $\mathrm{P}_{\mathrm{CA}} = \mathrm{P}_{\mathrm{CA}}(\mathbf{V}, \mathrm{O})$*, let* $\mathrm{DAG}(\mathbf{V})$ *be the set of DAGs on the fixed vertex set* $\mathbf{V}$*, and let* $G \models T$ *abbreviate* $G \models f$ *for every* $f \in T$*. For every* $T \subseteq \mathrm{O}$ *of valid CI facts,*

$$\{\pi_G(M) \mid M \in AS(\mathsf{enforce}(\mathrm{P}_{\mathrm{CA}}, T, \mathrm{O}))\} = \{G \in \mathrm{DAG}(\mathbf{V}) \mid G \models T\}.$$

In OptABA-PC, the objective atoms in O are the CI facts produced by MPC (Colombo & Maathuis, 2014). Let W contain their strictly positive release costs. OptABA-PC solves $\mathsf{opt}(\mathrm{P}_{\mathrm{CA}}, \mathrm{O}, \mathrm{W})$ with *clingo* (Gebser et al., 2011). Unlike ABA-PC's incremental release by strength, one solve minimises the total cost of the released set. Note that we use $w_f = \max\{1, \mathrm{round}(9{,}999{,}999 s(p_f))\}$, where $s(p) = 1 - p/(2\alpha)$ for $p < \alpha$ and $s(p) = (p + 1 - 2\alpha)/(2(1-\alpha))$ otherwise, following ABA-PC without weighting on conditioning set size. These costs encode preferences in line with the expected behaviour of p-values of being most uncertain around the threshold α (Hung et al., 1997; Casella & Berger, 2002), not calibrated probabilities.

Example 6 (OptABA-PC)**.** *Continuing Ex. 3 with the same fact weights, OptABA-PC releases the optimal MCS* $R = \{\texttt{indep(e,o,\{i\})}, \texttt{indep(r,i,\{e\})}, \texttt{indep(o,i,\{\})}\}$*. All three facts are wrong with respect to the* d*-separations of the DAG in Fig. 1; the repair retains all nine correct facts and the two remaining wrong facts. Its single compatible DAG is* $E \to O$*,* $R \to O$*,* $O \to I$*, with all three edges correctly oriented; while ABA-PC only one (cf. Ex. 3). Thus global minimum-cost repair can improve both CI selection and its graph witness.*

5.2 Optimality and Contestability Guarantees

Theorem 2 (OptABA-PC guarantees)**.** *Let* $\mathbf{V}$ *be finite,* O *a finite set of valid CI facts over* $\mathbf{V}$*, and* W *assign strictly positive release weights to every fact in* O*. Then* $\mathsf{opt}(\mathrm{P}_{\mathrm{CA}}, \mathrm{O}, \mathrm{W})$ *has an optimal stable model. For every such model* M*, put* $R = \rho(M)$ *and* $T = \mathrm{O} \setminus R$*. Then:*

(1) $\pi_G(M)$ *is a DAG on* $\mathbf{V}$ *satisfying every fact in* T*;*

(2) R *is an MCS with minimum release cost among all correction sets of* P_{CA} *w.r.t.* O*;*

(3) *for every* $f \in R$*, enforcing* $T \cup \{f\}$ *is incoherent, and a MUS* U_f *satisfies* $U_f \cap R = \{f\}$*.*

Conversely, every optimal MCS R *is* $\rho(M)$ *for some optimal stable model* M*, with the same cost.*

The correspondence is under projection, not a bijection: several optimal stable models may have the same released-fact projection and possibly different graph projections. These guarantees support two forms of user contestation. A release-cost contestation increases the penalty for releasing f; a fact-enforcement contestation excludes every repair that releases f.

Given a stable model M and a repair $R = \rho(M)$, a fact f is *retained* if $f \in M$, equivalently $f \notin R$, and *released* if $f \notin M$, equivalently $f \in R$. For a base program P, write $\mathcal{F} = CS(\mathrm{P}, \mathrm{O})$. In OptABA-PC, returned repairs are the projections $\rho(M)$ of optimal stable models. For a contested fact f, let the hard-retention program be:

$$\mathsf{opt}_f(\mathrm{P}, \mathrm{O}, \mathrm{W}) = \mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W}) \cup \{\texttt{:- not } f\}.$$

Property 1 (Release-cost contestability)**.** *For a weighted repair method over* $\mathcal{F}$*, let* $\mathrm{Rep}_{\mathrm{W}} \subseteq \mathcal{F}$ *denote the repairs returned under weights* W*. Suppose* W' *differs from* W *only in that* $w'_f > w_f$*. The method is* release-cost contestable at f *if its returned repairs can change only towards retaining* f*:*

$$\{R \in \mathrm{Rep}_{\mathrm{W}} : f \notin R\} \subseteq \mathrm{Rep}_{\mathrm{W}'}, \qquad \{R \in \mathrm{Rep}_{\mathrm{W}'} : f \in R\} \subseteq \mathrm{Rep}_{\mathrm{W}}\,.$$

Effectively, the optimal MCSes not containing f will remain optimal under W' while the ones that contain f under the updated W' must have already been optimal. Thus increasing the release cost may remove releasing repairs or add retaining repairs, but not conversely.

Property 2 (Fact-enforcement contestability)**.** *An ASP repair method based on* $\mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W})$ *is* fact-enforcement contestable at f *if*

$$\{\rho(M) \mid M \text{ is optimal for } \mathsf{opt}_f(\mathrm{P}, \mathrm{O}, \mathrm{W})\} = \arg\min_{R \in \mathcal{F},\, f \notin R} \mathcal{C}(R, \mathrm{W}).$$

It is total *on a fact domain if, for every fact* f *in that domain, some* $R \in \mathcal{F}$ *satisfies* $f \notin R$*. Thus making* f *mandatory returns exactly the best repairs that retain it.*

Proposition 2 (OptABA-PC contestability)**.** *Let* $\mathbf{V}$ *and* O *be finite, let every member of* O *be a valid CI fact over* $\mathbf{V}$*, put* $\mathrm{P}_{\mathrm{CA}} = \mathrm{P}_{\mathrm{CA}}(\mathbf{V}, \mathrm{O})$ *as in Equation* (1)*, and let* $\mathrm{W} = \{(o, w_o) \mid o \in \mathrm{O},\ w_o > 0\}$*. Then OptABA-PC satisfies Properties 1 and 2 for every* $f \in \mathrm{O}$*. Moreover, fact-enforcement contestability is total:* $\mathsf{opt}_f(\mathrm{P}_{\mathrm{CA}}, \mathrm{O}, \mathrm{W})$ *has an optimal stable model for every* $f \in \mathrm{O}$*.*

Besides making f mandatory, a user may increase its release cost, as in (Freedman et al., 2025). Property 1 guarantees that such graded contestation can only favour retention.

Definition 4 (Contestability margin)**.** *Let $\mathcal{F}$ be a finite family of repairs and* W *assign a finite release cost to every objective atom. For any fact f for which $\mathcal{F}$ contains both a repair retaining f and a repair releasing f, define*

$$c_{\mathrm{keep}}(f) = \min_{R \in \mathcal{F},\, f \notin R} \mathcal{C}(R, \mathrm{W}), \qquad c_{\mathrm{rel}}(f) = \min_{R \in \mathcal{F},\, f \in R} \mathcal{C}(R, \mathrm{W}).$$

If at least one globally optimal repair under W *releases f, then $c_{\mathrm{rel}}(f) \leq c_{\mathrm{keep}}(f)$. The* contestability margin *of f is $\mu(f) = c_{\mathrm{keep}}(f) - c_{\mathrm{rel}}(f) \geq 0$.*

Both minima exist: Prop. 2 gives a repair retaining f, and releasing all facts is feasible. Increasing w_f by δ adds δ only to repairs releasing f. Thus below $\mu(f)$ every optimum releases f, at equality both kinds coexist, and above it every optimum retains f; for integer weights, the least forcing increase is $\mu(f) + 1$ (proof in App. F, Prop. S.1).

Example 7 (Contestation of a released fact)**.** *Consider $f =$ `indep(e,o,{i})`, released in Ex. 6. For this illustration, set each release cost to $1000S$ using the strengths in Ex. 3. The three released facts in R therefore cost $c_{\mathrm{rel}}(f) = 1000 + 530 + 510 = 2040$. One MUS for f contains two retained facts: `dep(e,o,{})` and `indep(e,i,{o})`. The two independences imply `indep(e,o,{})`, contradicting the retained dependence; hence f cannot be reinstated while both other facts stay. The cheapest repair of the full trace with f mandatory replaces its release by releases of `indep(e,i,{o})` (cost 530) and `indep(e,r,{})` (cost 500), leaving the other two releases unchanged. Thus $c_{\mathrm{keep}}(f) = 2040 - 1000 + 530 + 500 = 2070$ and $\mu(f) = 30$. Increasing only w_f from 1000 to 1030 ties the two repair costs; 1031 makes every optimum retain f. The returned DAGs satisfy the requested f, even though, in this case, it is false wrt the generating DAG.*

Within a fixed trace and weighting scheme, the margin measures relative weighted degree of inconsistency; Apps. F and G.3 develop and evaluate this interpretation.

6 Experiments

Setup. Following (Russo et al., 2024), we use the Cancer, Earthquake, Survey, and Asia bnlearn networks from (Scutari, 2014) and add synthetic Erdős–Rényi (ER) (Erdős & Rényi, 1959) and scale-free (SF) (Barabási & Albert, 1999) DAGs with five and eight nodes. Each dataset has $N = 5000$ observations and 50 seeds; fixed networks retain their graph and conditional probability tables, whereas each synthetic seed generates both. ABA-PC and OptABA-PC use the same Bayes-ball encoding and receive identical MPC (Colombo & Maathuis, 2014) G^2 CI-test traces and weights ($\alpha = 0.01$), isolating their repair mechanisms. Fast Greedy Search (FGS) (Ramsey et al., 2017) and ASPCR-DAG (Hyttinen et al., 2014) use their native configurations. ABA runs use a 300-second solve limit and a separate 120-second graph-evaluation limit; $32/400$ OptABA-PC runs reach the optimisation limit and return a valid incumbent without proved optimality. Full protocol in App. G.2.

Metrics. Table 1 reports mean paired OptABA-PC–ABA-PC differences. Retained true facts agree with the generating DAG; FactF1 also penalises retained false and released true facts. #CPDAG counts saved equivalence classes satisfying all retained facts; F1 and Structural Hamming Distance (SHD) measure balance between precision and recall and number of mis-labled edges, respectively, against the reference CPDAG over these budget-limited sets. To evaluate contestability, we instantiate Def. 4 for every released fact f: the baseline optimum gives $c_{\mathrm{rel}}(f)$, a re-solve enforcing f gives $c_{\mathrm{keep}}(f)$, and their difference is the empirical margin $\mu(f)$. App. G gives metrics details alongside additional results (including Earthquake), absolute values, variability, significance, precision–recall, hard-retention feasibility, and margin distributions.

Results. Across all seven datasets in Table 1, OptABA-PC retains more true facts, identifies fewer compatible classes, and improves mean CPDAG F1 and SHD compared to ABA-PC. The main gains in Table 1 are stronger fact selection ($\Delta n^{T}_{\mathrm{acc}}$ and $\Delta\mathrm{FactF1}$) and fewer competing Markov Equivalence Classes ($\Delta\#\mathrm{CPDAG}$). Relative to ABA-PC, OptABA-PC retains more true independence and dependence facts on all seven datasets. FactF1 also improves on Asia and every synthetic benchmark, indicating better retain/release classification overall. Survey trades precision for more true-fact retention; Cancer's small FactF1 decline is not significant. Over 94% fewer compatible

Table 1: Paired OptABA-PC–ABA-PC effects on seven bnlearn and synthetic datasets (nodes in brackets). Entries are mean within-seed differences Δ; parentheses give relative change from ABA-PC, except runtime ($t_{\mathrm{Opt}}/t_{\mathrm{ABA}}$). Arrows indicate preferred directions. $\Delta n_{\mathrm{acc}}^{T}$ counts additional true CI facts retained; #CPDAG counts saved MECs satisfying every retained fact. Counts and graph means describe saved, budget-limited sets; Earthquake is omitted because outcomes do not differ. Tables 9 and 10 in the supplement give absolute mean $\pm$ standard deviation and significance for all datasets. Table 11 shows the CPDAG evaluation against the additional baselines.

CPDAG evaluation

Dataset	Δ#CPDAG $\downarrow$	ΔF1 $\uparrow$	ΔSHD $\downarrow$
Cancer (5)	-14.1 (-93.4%)	$+0.054$ ($+10.3\%$)	-0.974 (-34.6%)
Survey (6)	-1277.2 (-99.9%)	$+0.159$ ($+36.3\%$)	-2.790 (-39.1%)
Asia (8)	-10821.5 (-100.0%)	$+0.313$ ($+139.7\%$)	-10.507 (-73.1%)
ER (5)	-5.2 (-63.6%)	$+0.027$ ($+4.9\%$)	-0.792 (-27.5%)
ER (8)	-3835.3 (-99.9%)	$+0.150$ ($+31.3\%$)	-4.661 (-56.9%)
SF (5)	-3.3 (-3.7%)	$+0.067$ ($+10.1\%$)	-0.741 (-26.8%)
SF (8)	-2937.8 (-94.5%)	$+0.131$ ($+26.3\%$)	-3.725 (-45.7%)

Fact selection and runtime

Dataset	$\Delta n_{\mathrm{acc}}^{T}$ $\uparrow$	ΔFactF1 $\uparrow$	Δt $\downarrow$
Cancer (5)	$+1.6$ ($+6.8\%$)	-0.004 (-0.5%)	-0.233 ($\times 0.85$)
Survey (6)	$+4.9$ ($+19.6\%$)	-0.047 (-5.9%)	-12.728 ($\times 0.19$)
Asia (8)	$+21.8$ ($+20.6\%$)	$+0.082$ ($+9.2\%$)	-28.904 ($\times 0.70$)
ER (5)	$+1.3$ ($+4.1\%$)	$+0.016$ ($+1.6\%$)	-0.106 ($\times 0.90$)
ER (8)	$+15.4$ ($+21.5\%$)	$+0.082$ ($+9.5\%$)	$+91.044$ ($\times 2.90$)
SF (5)	$+1.7$ ($+6.0\%$)	$+0.034$ ($+3.6\%$)	-0.094 ($\times 0.95$)
SF (8)	$+21.1$ ($+29.2\%$)	$+0.093$ ($+11.1\%$)	$+56.807$ ($\times 2.34$)

CPDAGs are saved on every eight-variable benchmark, reducing uncertainty over structures consistent with the retained evidence. Against other baselines, graph accuracy is mixed: MPC and FGS often score better, particularly on synthetic benchmarks (Table 11). Our scores average over a returned set of compatible CPDAGs, whereas MPC and FGS return one. MPC has no significant advantage over the best F1 and SHD in our set on three of four synthetic settings (Tables 14 and 15). OptABA-PC achieves higher CPDAG and DAG F1 than its closest ASP-based counterpart, ASPCR-DAG, on all shared benchmarks.

CI–DAG correspondence further distinguishes the outputs: every graph by OptABA-PC satisfies the explicitly retained facts, with conflict certificates explaining releases, while MPC and ASPCR-DAG outputs respect their full CI traces in only $80/400$ (20%) and $32/250$ (12.8%) runs, respectively (Table 5). On datasets with at most six variables, all 742 requests to reinstate released facts yielded minimum-cost consistent repairs for OptABA-PC. For graded contestation, the median increase in a fact's release cost needed to make retention optimal was 45.5% (Tables 6 and 8). OptABA-PC uses one exact optimisation instead of repeated repair solves: it is faster than ABA-PC on five datasets, but slower on ER(8) and SF(8) (Table 1). Under matched heuristic repair, our Bayes-ball encoding is over $2\times$ as fast as the original active-path encoding on six- and eight-variable benchmarks (Table 4).

7 Conclusion

We cast causal discovery from noisy data as transparent CI fact repair. By aligning optimal MCSes with weak-constraint optimisation, OptABA-PC couples minimum cost CI fact repair with guaranteed CI–DAG correspondence, conflict explanations, and predictable redress. The contribution is optimal and provable redress, not solely competitive graph accuracy. OptABA-PC traceable link from retained facts to every returned DAG supports transparency and contestation. The empirical evidence of higher true CI retention also supports the effectiveness of the approach in contexts where data is limited or uncertain and external input is needed and available, making it particularly suited to high-stakes decisions with modest variable sets. The observed benefits and theoretical guarantees motivate more efficient methods that preserve these properties. Our proposed Bayes-ball encoding is one concrete step in this direction; scalability and robustness to release costs remain priorities.

AI USE STATEMENT

Generative AI tools were used to assist with editing and clarifying the exposition and with adapting the manuscript's formatting.

REPRODUCIBILITY STATEMENT

We provide the full source code in an accompanying archive. App. G.2 documents data generation and preprocessing, random seeds, method settings, computational limits, software versions, hardware, and evaluation procedures. App. G.3 details the contestability audits, and App. G.4 reports the full results. The encodings, formal assumptions, and proofs appear in Apps. B–F.

ETHICS STATEMENT

Contestability supports scrutiny of causal models, but it must be guarded against misuse: users could force unsupported CI claims or change release costs to favour a desired conclusion. Correspondence guarantees consistency with the retained facts, whose truth is not guaranteed (Ex. 7). Use in consequential decisions should therefore require documented reasons for contestations, an auditable record of the original evidence and all changes, and independent domain review. The resulting graphs and margins should inform human judgement alongside domain evidence, with attention to possible harms to affected people.

REFERENCES

Mario Alviano, Carmine Dodaro, Salvatore Fiorentino, Alessandro Previti, and Francesco Ricca. ASP and subset minimality: Enumeration, cautious reasoning and muses. *Artificial Intelligence*, 320:103931, 2023. ISSN 0004-3702. doi: 10.1016/j.artint.2023.103931.

Edoardo Amaldi, Marc E. Pfetsch, and Leslie E. Jr. Trotter. On the maximum feasible subsystem problem, IISs and IIS-hypergraphs. *Mathematical Programming*, 95(3):533–554, 2003. doi: 10.1007/s10107-002-0359-y.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286 (5439):509–512, 1999. URL http://www.doi.org/10.1126/science.286.5439.509.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995. doi: 10.1111/j.2517-6161.1995.tb02031.x.

Andrei Bondarenko, Phan Minh Dung, Robert A. Kowalski, and Francesca Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93:63–101, 1997. doi: 10.1016/S0004-3702(97)00015-5.

Giorgos Borboudakis and Ioannis Tsamardinos. Towards robust and versatile causal discovery for business applications. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016*, pp. 1435–1444, 2016. doi: 10.1145/2939672.2939872.

Francesco Calimeri, Wolfgang Faber, Martin Gebser, Giovambattista Ianni, Roland Kaminski, Thomas Krennwallner, Nicola Leone, Marco Maratea, Francesco Ricca, and Torsten Schaub. Asp-core-2 input language format. *Theory and Practice of Logic Programming*, 20(2):294–309, 2020. doi: 10.1017/S1471068419000450. URL https://doi.org/10.1017/S1471068419000450.

G. Casella and R.L. Berger. *Statistical Inference*. Thomson Learning, 2002. ISBN 9780534243128. doi: 10.1201/9781003456285.

David Maxwell Chickering. Learning equivalence classes of bayesian-network structures. *Journal of Machine Learning Research*, 2:445–498, 2002a. URL https://jmlr.org/papers/v2/chickering02a.html.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002b. URL https://jmlr.org/papers/v3/chickering02b.html.

John W. Chinneck and E. W. Dravnieks. Locating minimal infeasible constraint sets in linear programs. *ORSA Journal on Computing*, 3(2):157–168, 1991. doi: 10.1287/ijoc.3.2.157.

Diego Colombo and Marloes H. Maathuis. Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 15(1):3741–3782, 2014. doi: 10.5555/2627435.2750365.

Kristijonas Cyras, Xiuyi Fan, Claudia Schulz, and Francesca Toni. Assumption-based argumentation: Disputes, explanations, preferences. *IfCoLog Journal of Logics and their Applications*, 4(8), 2017. URL http://www.collegepublications.co.uk/downloads/ifcolog00017.pdf.

Paul Erdős and Alfréd Rényi. On random graphs I. *Publicationes Mathematicae Debrecen*, 6(290-297):18, 1959.

Sofia Faltenbacher, Jonas Wahl, Rebecca Jean Herman, and Jakob Runge. How PC-based methods err: Towards better reporting of assumption violations and small sample errors. In Emilija Perković and Daniel Malinsky (eds.), *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence*, volume 337 of *Proceedings of Machine Learning Research*, pp. 1498–1519. PMLR, 17–21 Aug 2026. URL https://proceedings.mlr.press/v337/faltenbacher26a.html.

Gabriel Freedman, Adam Dejl, Deniz Gorur, Xiang Yin, Antonio Rago, and Francesca Toni. Argumentative large language models for explainable and contestable claim verification. *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI 2025*, 39(14):14930–14939, 2025. doi: 10.1609/aaai.v39i14.33637. URL https://doi.org/10.1609/aaai.v39i14.33637.

Martin Gebser, Roland Kaminski, and Torsten Schaub. Complex optimization in answer set programming. *Theory and Practice of Logic Programming*, 11(4-5):821–839, 2011. doi: 10.1017/S1471068411000329.

Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Multi-shot ASP solving with clingo. *Theory Pract. Log. Program.*, 19(1):27–82, 2019. URL https://doi.org/10.1017/S1471068418000054.

Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In *Proceedings of the International Logic Programming Conference and Symposium*, pp. 1070–1080. MIT Press, 1988. URL https://www.cs.utexas.edu/~ai-lab/pubs/gel88.pdf.

Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, 10:524, 2019. doi: 10.3389/fgene.2019.00524.

H. M. James Hung, Robert T. O'Neill, Peter Bauer, and Karl Kohne. The behavior of the p-value when the alternative hypothesis is true. *Biometrics*, 53(1):11–22, 1997. ISSN 0006341X, 15410420. URL http://www.jstor.org/stable/2533093.

Antti Hyttinen, Patrik O. Hoyer, Frederick Eberhardt, and Matti Järvisalo. Discovering cyclic causal models with latent variables: A general sat-based procedure. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, UAI 2013*, 2013. URL https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=2391&proceeding_id=29.

Antti Hyttinen, Frederick Eberhardt, and Matti Järvisalo. Constraint-based Causal Discovery: Conflict Resolution with Answer Set Programming. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, UAI 2014*, pp. 340–349, 2014. URL https://www.cs.helsinki.fi/u/mjarvisa/papers/hyttinen-eberhardt-jarvisalo.uai14.pdf.

Antti Hyttinen, Paul Saikko, and Matti Järvisalo. A core-guided approach to learning optimal causal graphs. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017*, pp. 645–651, 2017. doi: 10.24963/ijcai.2017/90.

Francesco Leofante, Hamed Ayoobi, Adam Dejl, Gabriel Freedman, Deniz Gorur, Junqi Jiang, Guilherme Paulino-Passos, Antonio Rago, Anna Rapberger, Fabrizio Russo, Xiang Yin, Dekai Zhang, and Francesca Toni. Contestable AI needs computational argumentation. In *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning, KR 2024*, 2024. URL https://doi.org/10.24963/kr.2024/83.

Mark H. Liffiton and Karem A. Sakallah. Algorithms for computing minimal unsatisfiable subsets of constraints. *Journal of Automated Reasoning*, 40(1):1–33, 2008. doi: 10.1007/s10817-007-9084-z.

Vladimir Lifschitz. *Answer Set Programming*. Springer, Cham, 2019. ISBN 978-3-030-24658-7. doi: 10.1007/978-3-030-24658-7.

Sara Magliacane, Tom Claassen, and Joris M. Mooij. Ancestral causal inference. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems, NeurIPS 2016*, pp. 4466–4474, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/f3d9de86462c28781cbe5c47ef22c3e5-Abstract.html.

Erick Moreno-Centeno and Richard M. Karp. The implicit hitting set approach to solve combinatorial optimization problems with an application to multigenome alignment. *Operations Research*, 61(2):453–468, 2013. doi: 10.1287/opre.1120.1139.

António Morgado, Federico Heras, Mark H. Liffiton, Jordi Planes, and João Marques-Silva. Iterative and core-guided maxsat solving: A survey and assessment. *Constraints An Int. J.*, 18(4):478–534, 2013. doi: 10.1007/S10601-013-9146-2. URL https://doi.org/10.1007/s10601-013-9146-2.

Judea Pearl. *Probabilistic reasoning in intelligent systems - networks of plausible inference*. Morgan Kaufmann, 1989. ISBN 978-0-08-051489-5. doi: 10.1016/C2009-0-27609-4.

Judea Pearl. *Causality*. Cambridge University Press, 2 edition, 2009. doi: 10.1017/CBO9780511803161.

Joseph D. Ramsey, Madelyn Glymour, Ruben Sanchez-Romero, and Clark Glymour. A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *Int. J. Data Sci. Anal.*, 3(2):121–129, 2017. URL https://doi.org/10.1007/s41060-016-0032-z.

Kari Rantanen, Antti Hyttinen, and Matti Järvisalo. Discovering causal graphs with cycles and latent confounders: An exact branch-and-bound approach. *Int. J. Approx. Reason.*, 117:29–49, 2020. doi: 10.1016/j.ijar.2019.10.009.

Anna Rapberger, Markus Ulbricht, and Francesca Toni. On the correspondence of non-flat assumption-based argumentation and logic programming with negation as failure in the head. In *Proceedings of the 22nd International Workshop on Nonmonotonic Reasoning (NMR 2024)*, volume 3835 of *CEUR Workshop Proceedings*, pp. 112–121, 2024. URL https://ceur-ws.org/Vol-3835/paper12.pdf.

Raymond Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, 1987. doi: 10.1016/0004-3702(87)90062-2.

Thomas S. Richardson and Peter Spirtes. Ancestral graph markov models. *The Annals of Statistics*, 30(4):962–1030, 2002. doi: 10.1214/aos/1031689015.

Fabrizio Russo, Anna Rapberger, and Francesca Toni. Argumentative causal discovery. In *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning, KR 2024*, pp. 938–949, 2024. doi: 10.24963/KR.2024/88.

Marco Scutari. Bayesian network repository. http://www.bnlearn.com/bnrepository, 2014.

Ross D. Shachter. Bayes-ball: Rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, UAI 1998*, pp. 480–487. Morgan Kaufmann, 1998.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search, Second Edition*. Adaptive computation and machine learning. MIT Press, 2000. ISBN 978-0-262-19440-2. URL https://www.cs.cmu.edu/afs/andrew/scs/cs/15-381/archive/OldFiles/lib/cvsub/.g/group/sdss/.g/group2/g/scottd/fullbook.pdf.

Sofia Triantafillou and Ioannis Tsamardinos. Constraint-based causal discovery from multiple interventions over overlapping variable sets. *Journal of Machine Learning Research*, 16(66):2147–2205, 2015. URL http://jmlr.org/papers/v16/triantafillou15a.html.

Sofia Triantafilou, Ioannis Tsamardinos, and Ioannis G. Tollis. Learning causal structure from overlapping variable sets. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010*, volume 9, pp. 860–867, 2010. URL http://proceedings.mlr.press/v9/triantafillou10a.html.

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. doi: 10.1007/s10994-006-6889-7.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945. doi: 10.2307/3001968.

S. S. Wilks. The likelihood test of independence in contingency tables. *The Annals of Mathematical Statistics*, 6(4):190–196, 1935. doi: 10.1214/aoms/1177732564.

Alessio Zanga, Elif Ozkirimli, and Fabio Stella. A survey on causal discovery: Theory and practice. *Int. J. Approx. Reason.*, 151:101–129, 2022. URL https://doi.org/10.1016/j.ijar.2022.09.004.

Zhalama, Jiji Zhang, Frederick Eberhardt, and Wolfgang Mayer. Sat-based causal discovery under weaker assumptions. *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017*, 2017. URL http://auai.org/uai2017/proceedings/papers/234.pdf.

Zhalama, Jiji Zhang, Frederick Eberhardt, Wolfgang Mayer, and Mark Junjie Li. Asp-based discovery of semi-markovian causal models under weaker assumptions. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*, pp. 1488–1494, 2019. URL https://doi.org/10.24963/ijcai.2019/206.

A ASP SEMANTICS

We make explicit the semantics used in the main paper. A finite ground normal rule is

$$h \texttt{ :- } a_1, \ldots, a_m, \texttt{not } b_1, \ldots, \texttt{not } b_n,$$

where the head h may be absent. Its body is true in an interpretation I iff $a_i \in I$ for every i and $b_j \notin I$ for every j. The rule is satisfied iff its body is false or $h \in I$; when the head is absent, satisfaction therefore requires the body to be false.

For a program P and interpretation I, the Gelfond–Lifschitz reduct P^I deletes each rule having some negative body literal $\texttt{not } b$ with $b \in I$ and removes all remaining negative body literals (Gelfond & Lifschitz, 1988). Then I is a stable model of P iff it is a subset-minimal model of P^I. Non-ground safe programs are interpreted through their finite ground instantiation. The encoding also uses ground choice and disjunctive constructs, including the inherited default-negated head literal in Listing 2, under their *clingo* stable-model semantics (Gebser et al., 2011; Calimeri et al., 2020).

A ground weak constraint $\texttt{:}\sim B.\ [w@l, \vec{t}]$ is active in I when B is true in I. At priority l, the solver sums w once for each distinct activated tuple $(w, l, \vec{t})$ and minimises the resulting cost vectors lexicographically by priority. OptABA-PC uses

$$\texttt{:}\sim \texttt{ not } o.\ [w_o@1, o]$$

for every objective atom o. Including o as a tuple identifier is essential: two different facts of equal weight then activate different tuples and are both counted. All release weights are strictly positive.

B CAUSAL ABA ENCODINGS

Equation (1) separates the rules inherited unchanged from Causal ABA (Russo et al., 2024), given in Listings 1–3, from the new Bayes-ball encoding in Listing 4. This section gives every rule schema in the inherited component $\mathrm{P}_{\mathrm{core}}^{\mathrm{CABA}}$. Together with Listing 4, Listings 1–3 completely define P_{CA}.

B.1 ORIGINAL CAUSAL ABA ENCODINGS

The instance supplies one fact `var(v)` for every $v \in \mathbf{V}$ and one fact `in(v,s)` for every member v of each represented conditioning set $s \in \mathcal{S}_{\mathrm{O}}$; `empty` denotes $\emptyset$. A CI fact is *valid over* $\mathbf{V}$ when it asserts independence or dependence for distinct endpoints $X, Y \in \mathbf{V}$ given $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}$. In Listing 1, lines 1–2 construct the conditioning-set domain, lines 3–4 close CI atoms under endpoint symmetry, and line 5 rejects a simultaneous independence and dependence assertion for the same valid query.

```
1 set(S) :- in(N,S).
2 set(empty).
3 indep(X,Y,S) :- indep(Y,X,S), var(X), var(Y), set(S), X!=Y.
4 dep(X,Y,S) :- dep(Y,X,S), var(X), var(Y), set(S), X!=Y.
5 :- dep(X,Y,S), indep(X,Y,S), set(S), var(X), var(Y), X!=Y, not in(X,S),
       not in(Y,S).
```

Listing 1: Inherited conditioning-set and CI-consistency rules.

The candidate-graph rules in Listing 2 are also inherited unchanged. Line 1 chooses a directed edge in either orientation or no adjacency. Lines 2–3 make `edge/2` the symmetric adjacency relation. Line 4 exploits the fact that a direct edge cannot coexist with a valid conditional independence between its endpoints (Spirtes et al., 2000); lines 5–7 keep edge and arrow choices consistent and exclude a two-cycle.

```
1 arrow(X,Y) | arrow(Y,X) | not edge(X,Y) :- var(X), var(Y), X!=Y.
2 edge(X,Y) :- arrow(X,Y), X!=Y, var(X), var(Y).
3 edge(X,Y) :- edge(Y,X), X!=Y, var(X), var(Y).
4 :- edge(X,Y), indep(X,Y,S), set(S), var(X), var(Y), X!=Y, not in(X,S),
     not in(Y,S).
5 :- arrow(X,Y), not edge(X,Y), var(X), var(Y).
6 :- edge(X,Y), not arrow(X,Y), not arrow(Y,X), var(X), var(Y).
7 :- arrow(Y,X), arrow(X,Y), X!=Y, var(X), var(Y).
```

Listing 2: Inherited candidate-DAG and adjacency rules.

Finally, lines 1–2 of Listing 3 define directed reachability, and line 3 excludes every directed cycle.

```
1 dpath(X,Y) :- arrow(X,Y), X!=Y, var(X), var(Y).
2 dpath(X,Y) :- arrow(X,Z), dpath(Z,Y), var(X).
3 :- dpath(X,X), var(X).
```

Listing 3: Inherited reachability and acyclicity rules.

Before any CI fact is enforced, the graph projections produced by Listings 2–3 are exactly all labelled DAGs on the fixed vertex set $\mathbf{V}$; enforcing CI facts subsequently filters this family. The original Causal ABA encoding continues with instance-generated simple-path witnesses and `collider`/`nb` rules. Those path-enumerating rules are precisely the component replaced by $\mathrm{P}_{\mathrm{BB}}^{\mathrm{new}}$. They are therefore not included in $\mathrm{P}_{\mathrm{core}}^{\mathrm{CABA}}$ or in P_{CA}; $\mathrm{P}_{\mathrm{BB}}^{\mathrm{new}}$ derives `ap3/3` without materialising paths and enforces the selected `dep`/`indep` atoms against that result. Thus Listings 1–3, together with Listing 4, display every rule affecting the graph, Bayes-ball closure, or CI enforcement.

B.2 Novel Bayes-ball encoding

We instantiate Bayes-ball for the input CI queries and their symmetric counterparts: for each objective atom over $(X, Y, \mathbf{Z})$, the instance supplies both `qpair(X,Y,Z)` and `qpair(Y,X,Z)`, independently of whether that atom is selected. This symmetric query domain ensures that the enforcement constraints also check the reversed CI atoms derived by Listing 1. The facts `in(N,Z)` represent membership $N \in \mathbf{Z}$. Listing 4 gives the complete Bayes-ball encoding. Lines 2–4 compute whether a node belongs to, or has a descendant in, $\mathbf{Z}$, and whether it has at least two distinct parents. The predicate `is_collider` tests only this two-parent condition; collider status on a particular path remains path-relative. The state `bb_u(X,N,Z)` records arrival at N from a child of N; `bb_d(X,N,Z)` records arrival from a parent. Lines 5–6 initialise both states at the source.

```
1 qpair(X,Y,Z). % generated for queried CI facts
2 anc_obs(N,Z) :- in(N,Z).
3 anc_obs(N,Z) :- dpath(N,M), in(M,Z).
4 is_collider(N) :- arrow(P,N), arrow(Q,N), P<Q.
5 bb_u(X,X,Z) :- qpair(X,_,Z).
6 bb_d(X,X,Z) :- qpair(X,_,Z).
7 bb_u(X,P,Z) :- bb_u(X,N,Z), not in(N,Z), arrow(P,N).
8 bb_u(X,P,Z) :- bb_d(X,N,Z), is_collider(N), anc_obs(N,Z), arrow(P,N).
9 bb_d(X,C,Z) :- bb_u(X,N,Z), not in(N,Z), arrow(N,C).
10 bb_d(X,C,Z) :- bb_d(X,N,Z), not in(N,Z), arrow(N,C).
11 ap3(X,Y,Z) :- qpair(X,Y,Z), bb_u(X,Y,Z), not in(X,Z), not in(Y,Z).
12 ap3(X,Y,Z) :- qpair(X,Y,Z), bb_d(X,Y,Z), not in(X,Z), not in(Y,Z).
13 :- dep(X,Y,Z),   not ap3(X,Y,Z).
14 :- indep(X,Y,Z),     ap3(X,Y,Z).
```

Listing 4: Bayes-ball encoding $\mathrm{P}_{\mathrm{BB}}^{\mathrm{new}}$ of OptABA-PC.

The four propagation rules in Listing 4 distinguish both arrival direction and destination. Line 7 sends a ball that arrived from a child through an unobserved N to each parent of N, while line 9 sends the same state to each child. Line 8 sends a ball that arrived from a parent out through the parents of a collider that is itself in, or has a descendant in, $\mathbf{Z}$; this is the contracted opened-collider transition justified in Appendix C (Lemma S.2). Line 10 continues a parent-arriving ball to each child of an unobserved N. All other local cases are blocked and therefore have no propagation rule. Lines 11–12 derive `ap3` when either state reaches the target, witnessing d-connection for a valid

query. Line 13 requires this witness for an enforced dependence, while line 14 forbids it for an enforced independence, preserving the Causal ABA correspondence between selected CI facts and graph semantics.

C Correctness of the Implemented Bayes-Ball Rules

To prove the Bayes-ball correctness lemma recalled below, fix the objects in its statement: a finite DAG $G = (V, E)$ and a valid query $(X, Y, \mathbf{Z})$ with distinct endpoints $X, Y \notin \mathbf{Z}$. Let `Z` denote the ASP identifier encoding $\mathbf{Z}$, so that `in(v,Z)` holds exactly when $v \in \mathbf{Z}$. For any $A \subseteq V$, let

$$\mathrm{An}_G(A) := \{v \in V \mid v \rightsquigarrow_G a \text{ for some } a \in A\},$$

where $v \rightsquigarrow_G a$ denotes a directed path in G, including one of length zero. Thus $A \subseteq \mathrm{An}_G(A)$. Once G and $\mathbf{Z}$ are fixed, the recursive definitions of `dpath/2`, `anc_obs/2`, `bb_u/3`, `bb_d/3`, and `ap3/3` form finite positive strata and therefore have a least fixed point. Lines 2–3 of Listing 4 give

$$\texttt{anc_obs(v,Z)} \iff v \in \mathrm{An}_G(\mathbf{Z}),$$

and line 4 defines `is_collider`(v) to hold exactly when v has at least two distinct parents in G.

The atom `bb_u(X,v,Z)` records arrival at v from a child, and `bb_d(X,v,Z)` records arrival from a parent. With X and the identifier `Z` for $\mathbf{Z}$ fixed, write $u(v)$ for `bb_u(X,v,Z)` and $d(v)$ for `bb_d(X,v,Z)`. Lines 5–6 of Listing 4 initialise both states at X, and lines 7–10 close them under:

$$u(v),\ v \notin \mathbf{Z},\ p \to v \Longrightarrow u(p), \tag{2}$$

$$u(v),\ v \notin \mathbf{Z},\ v \to c \Longrightarrow d(c), \tag{3}$$

$$d(v),\ v \notin \mathbf{Z},\ v \to c \Longrightarrow d(c), \tag{4}$$

$$d(v),\ v \in \mathrm{An}_G(\mathbf{Z}),\ \texttt{is_collider}(v),\ p \to v \Longrightarrow u(p). \tag{5}$$

Equations (2), (3), (4), and (5) are, respectively, lines 7, 9, 10, and 8 of Listing 4.

Shachter's operational characterisation. A structured belief network in Shachter (1998) is (V, E, F), where F is the set of nodes whose variables are deterministic functions of their parents. Such a node need not be a root or a constant; $F = \emptyset$ here means that our DAG query supplies no functional-determinacy annotations. Apply Shachter's Algorithm 2 with source set $J = \{X\}$ and observed set $K = \mathbf{Z}$. Algorithm 2 schedules each node for processing according to whether the ball arrives from a child or a parent; sending a ball to its parents marks its top, while sending a ball to its children marks its bottom (Shachter, 1998, Alg. 2). These marks are not the ASP states themselves: $u(v)$ and $d(v)$ encode the two arrival directions. For example, at an unobserved v, line 7 of Listing 4 sends $u(v)$ to its parents and corresponds to a top mark, whereas lines 9–10 send $u(v)$ or $d(v)$ to its children and correspond to a bottom mark. Because $F = \emptyset$, either arrival direction at an unobserved node causes a bottom mark. Lemma S.1 states the specialisation of Shachter's irrelevance theorem needed below.

Lemma S.1 (Shachter specialisation)**.** *For every $v \notin \mathbf{Z}$, the execution of Shachter's Algorithm 2 marks v on the bottom iff there is an active path from X to v given $\mathbf{Z}$, equivalently iff X and v are d-connected given $\mathbf{Z}$.*

Proof. Apply Theorem 2 of Shachter (1998) with the set-valued parameters $J = \{X\}$, $K = \mathbf{Z}$, and $L = \{v\}$, and with $F = \emptyset$. Step 4 of Algorithm 2 defines $N_i(J \mid K)$ as exactly the nodes not marked on the bottom, while Theorem 2 identifies $L \subseteq N_i(J \mid K)$ with irrelevance of X_J to X_L given X_K. Hence v is bottom-marked iff it is not irrelevant to X given $\mathbf{Z}$. Shachter's Theorem 1 identifies irrelevance with the absence of an active path; when $F = \emptyset$, this is ordinary d-separation. Thus v is bottom-marked iff an active path exists, equivalently iff X and v are d-connected given $\mathbf{Z}$. □

The ASP rules are deliberately a collapsed version of Shachter's schedule. Lemma S.2 states the simulation property that must be proved for our implementation.

Lemma S.2 (ASP–Bayes-ball simulation)**.** *For every $v \notin \mathbf{Z}$, either $u(v)$ or $d(v)$ belongs to the ASP closure iff the execution of Shachter's Algorithm 2 marks v on the bottom.*

Proof. A state $u(v)$ represents Shachter's Algorithm 2 scheduling v as reached from a child, and $d(v)$ represents scheduling it as reached from a parent. At an unobserved probabilistic node, Steps 3(c)(i)–(ii) send a child-arriving ball to parents and children, and Step 3(d)(ii) sends a parent-arriving ball to children. These transitions are exactly Equations (2)–(4). The remaining ASP transition, Equation (5), contracts the detour used in the proof of Shachter's Theorem 2: from an ancestor n, Algorithm 2 can travel down a directed path to an observed $z \in \mathbf{Z}$, bounce there under Step 3(d)(i), return to n, and leave through a parent of n. Such a detour exists exactly when $n \in \mathrm{An}_G(\mathbf{Z})$.

($\Rightarrow$) First omit the additional seed $d(X)$ from line 6 of Listing 4. Because $X \notin \mathbf{Z}$, every propagation consequence of this seed already follows from $u(X)$: its child transitions are available from $u(X)$, as are any parent transitions. Omitting the seed therefore does not change which vertices are reached, and X itself remains reached by $u(X)$. For the remaining derivations, induction shows that each reached state can be scheduled by Shachter's Algorithm 2 with the corresponding arrival direction. The base state $u(X)$ matches Algorithm 2, Step 2. Each application of Equations (2)–(4) is an ordinary transition. Equation (5) expands into the directed detour to an observed descendant and back described above, using the first observed node on that path (or bouncing immediately if the current node is observed). Thus every ASP-reached unobserved vertex is scheduled and, since $F = \emptyset$, bottom-marked.

($\Leftarrow$) Suppose Shachter's Algorithm 2 bottom-marks an unobserved vertex v. If $v = X$, then $u(X)$ is already initial. Otherwise, Lemma S.1 supplies a simple active path from X to v. The initial state $u(X)$ can traverse its first edge in either direction because $X \notin \mathbf{Z}$. Follow the path inductively. At each non-collider, activity gives a vertex outside $\mathbf{Z}$, and the arrival and departure directions are exactly one of the three ordinary transitions in Equations (2)–(4). At each collider, the path enters and leaves along two distinct incoming edges, so `is_collider` holds; activity also places the collider in $\mathrm{An}_G(\mathbf{Z})$. Equation (5) therefore traverses that step. The final step derives either $u(v)$ or $d(v)$. This proves the required equivalence for reached vertices; it does not require the two procedures to generate identical arrival states. □

We now recall the Bayes-ball correctness lemma from the main paper.

Lemma 2 (Bayes-ball encoding correctness)**.** *Let $\mathbf{V}$ be finite, let $G \in \mathrm{DAG}(\mathbf{V})$ fix the `arrow` atoms, and let `qpair(X,Y,Z)` encode a valid query with distinct endpoints $X, Y \in \mathbf{V} \setminus \mathbf{Z}$. In the least fixed point of the grounded propagation rules in Listing 4, `ap3(X,Y,Z)` is derived iff X and Y are d-connected given $\mathbf{Z}$ in G.*

Proof. Because V is finite, grounding the propagation rules in Listing 4 for the fixed `arrow`, `in`, and `qpair(X,Y,Z)` atoms gives a finite program whose positive recursive strata have the least fixed point described above. The endpoint guards `not in(X,Z)` and `not in(Y,Z)` on lines 11–12 hold because validity gives $X, Y \notin \mathbf{Z}$.

($\Rightarrow$) If `ap3(X,Y,Z)` is derived, one of the endpoint rules on lines 11–12 of Listing 4 has reached Y in state $u(Y)$ or $d(Y)$. Lemma S.2 gives a bottom mark at Y in Shachter's Algorithm 2, and Lemma S.1 yields d-connection given $\mathbf{Z}$.

($\Leftarrow$) If X and Y are d-connected given $\mathbf{Z}$, Lemma S.1 gives a bottom mark at Y in Shachter's Algorithm 2. Lemma S.2 then derives $u(Y)$ or $d(Y)$, so the corresponding endpoint rule on line 11 or 12 of Listing 4 derives `ap3(X,Y,Z)`. □

Let P_{CA} be the finite ground program defined component-wise in Equation (1): the inherited core in Listings 1–3, together with the Bayes-ball encoding in Listing 4. The objective-atom choices and weak constraints are not part of P_{CA}; they are introduced by `enforce` and `opt`. Recall from the main paper that, for a stable model M of a Causal ABA instance on variables V, its *graph projection* is

$$\pi_G(M) = (V, E_M), \qquad E_M = \{(x, y) \in V^2 \mid \texttt{arrow}(x, y) \in M\}.$$

Unlike $\rho(M)$, which records released objective atoms, $\pi_G(M)$ discards all non-edge atoms.

We now recall the graph-fidelity corollary from the main paper.

Corollary 1 (Graph fidelity)**.** *For the finite sets $\mathbf{V}$ and O defining $\mathrm{P}_{\mathrm{CA}} = \mathrm{P}_{\mathrm{CA}}(\mathbf{V}, \mathrm{O})$, let $\mathrm{DAG}(\mathbf{V})$ be the set of DAGs on the fixed vertex set $\mathbf{V}$, and let $G \models T$ abbreviate $G \models f$ for every $f \in T$. For*

every $T \subseteq \mathrm{O}$ of valid CI facts,

$$\{\pi_G(M) \mid M \in AS(\mathsf{enforce}(\mathrm{P}_{\mathrm{CA}}, T, \mathrm{O}))\} = \{G \in \mathrm{DAG}(\mathbf{V}) \mid G \models T\}.$$

Proof. Fix the finite sets $\mathbf{V}$ and O and the program $\mathrm{P}_{\mathrm{CA}}(\mathbf{V}, \mathrm{O})$ from Equation (1).

($\subseteq$) Let $M \in AS(\mathsf{enforce}(\mathrm{P}_{\mathrm{CA}}, T, \mathrm{O}))$. Lines 1 and 5–7 of Listing 2 choose at most one orientation per unordered vertex pair, and line 3 of Listing 3 rejects every directed cycle; hence $\pi_G(M) \in \mathrm{DAG}(\mathbf{V})$. For every enforced dependence fact, line 13 of Listing 4 requires `ap3`; for every enforced independence fact, line 14 forbids `ap3`. Lemma 2 therefore gives $\pi_G(M) \models T$.

($\supseteq$) Fix a DAG G on $\mathbf{V}$ satisfying T. Proposition 3.3 of Russo et al. (2024) and the stable ABA–ASP correspondence (Rapberger et al., 2024) supply stable choices of the inherited graph generator whose `arrow` atoms are exactly the edges of G. Choose every objective atom in T and close these choices under CI symmetry; no atom outside this symmetric closure needs to be selected. The remaining finite positive or stratified rules have a least closure. Every derived CI fact is either in T or its endpoint reversal, so G satisfies it. The symmetric query domain ensures that Lemma 2 applies to each such fact in the enforcement constraints. Acyclicity holds because G is a DAG, and an enforced independence cannot join adjacent endpoints because a direct edge would be an active path. Thus all graph and CI constraints hold. Constraint filtering (6) preserves the resulting stable model after the constraints enforcing T are added, giving some $M \in AS(\mathsf{enforce}(\mathrm{P}_{\mathrm{CA}}, T, \mathrm{O}))$ with $\pi_G(M) = G$. □

D OPTIMAL-MCS

Example 8 (Optimal MCS)**.** *Let* P *and* O *be the program and objective atoms from Ex. 4, respectively. Then $MCS(\mathrm{P}, \mathrm{O}) = \{M_1, M_2, M_3\}$, with $M_1 = \{o(1), o(4)\}$, $M_2 = \{o(2), o(4)\}$, and $M_3 = \{o(3), o(4)\}$. Let $\mathrm{W} = \{(o(1), 5), (o(2), 10), (o(3), 15), (o(4), 20)\}$. Then $\mathcal{C}(M_1, \mathrm{W}) = 25$, $\mathcal{C}(M_2, \mathrm{W}) = 30$, and $\mathcal{C}(M_3, \mathrm{W}) = 35$, hence M_1 is the unique optimal MCS by Def. 3.*

Example 9 (MCSes and Weak Constraints)**.** *Let* P *and* O *be the program and objective atoms from Ex. 4, respectively, and let* W *be as in Ex. 8. Then* $\mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W})$ *is:*

```
{o(1);o(2);o(3);o(4)}.
a :- not b, o(1).  b :- not c, o(2).  c :- not a, o(3).  :- not d, o(4).
:~ not o(1). [5@1,o(1)]   :~ not o(2). [10@1,o(2)]
:~ not o(3). [15@1,o(3)]  :~ not o(4). [20@1,o(4)]
```

An optimal stable model of $\mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W})$ *is $M = \{o(2), o(3), c\}$, making $\mathrm{O} \setminus M = \{o(1), o(4)\}$ an optimal MCS for* P *w.r.t.* O, W.

D.1 PROOFS

Throughout this section, P is a finite ground program, O is a finite set of objective atoms, and $\mathrm{W} = \{(o, w_o) \mid o \in \mathrm{O},\ w_o > 0\}$ assigns one strictly positive release weight to every objective atom.

Let

$$A(\mathrm{P}, \mathrm{O}) = \mathrm{P} \cup \{\{o\}. \mid o \in \mathrm{O}\}.$$

For $S \subseteq \mathrm{O}$, $\mathsf{enforce}(\mathrm{P}, S, \mathrm{O})$ adds to $A(\mathrm{P}, \mathrm{O})$ the constraints `:- not` o for $o \in S$. For a stable model M of $A(\mathrm{P}, \mathrm{O})$, its *released-fact projection* is $\rho(M) = \mathrm{O} \setminus M$: it records exactly the objective atoms that are false in M.

We first recall the positive-cost lemma from the main paper.

Lemma 1 (Positive-cost minimality)**.** *Let* P *be a finite ground program, let* O *be finite, and let* $\mathrm{W} = \{(o, w_o) \mid o \in \mathrm{O},\ w_o > 0\}$. *Every minimum-cost member of $CS(\mathrm{P}, \mathrm{O})$ is an optimal MCS with respect to* W.

Proof. Let R be a minimum-cost member of $CS(\mathrm{P}, \mathrm{O})$. If a strict subset $R' \subset R$ were also a correction set, then $\mathcal{C}(R', \mathrm{W}) < \mathcal{C}(R, \mathrm{W})$ because every weight in the nonempty difference $R \setminus R'$

is strictly positive, contradicting the choice of R. Thus $R \in MCS(\mathrm{P}, \mathrm{O})$. Since $MCS(\mathrm{P}, \mathrm{O}) \subseteq CS(\mathrm{P}, \mathrm{O})$, the minimum cost of R over all correction sets is no greater than that of any MCS; hence R is an optimal MCS with respect to W. □

Consequently, whenever $CS(\mathrm{P}, \mathrm{O}) \neq \emptyset$, minimising over correction sets has the same value as minimising over MCSes.

We use the standard theorem that constraints filter, but do not create, stable models. For every finite ground program Q and finite set of constraints D,

$$AS(Q \cup D) = \{M \in AS(Q) \mid M \models D\}, \tag{6}$$

where $M \models D$ means that M satisfies every constraint in D (Lifschitz, 2019, Sec. 5.5, Theorem on Constraints).

Lemma S.3 (Exact MCS realisability)**.** *Let* P *be a finite ground program, let* O *be finite, and let* $\mathrm{W} = \{(o, w_o) \mid o \in \mathrm{O},\ w_o > 0\}$. *For every* $R \in MCS(\mathrm{P}, \mathrm{O})$ *there exists a stable model* M_R *of* $A(\mathrm{P}, \mathrm{O})$ *such that* $\rho(M_R) = R$. *The same* M_R *is a stable model of* $\mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W})$*, and its weak-constraint cost is exactly* $\mathcal{C}(R, \mathrm{W})$.

Proof. Because R is a correction set, $\mathsf{enforce}(\mathrm{P}, \mathrm{O} \setminus R, \mathrm{O})$ is coherent; choose one of its stable models M_R. This program is $A(\mathrm{P}, \mathrm{O})$ together with the constraints enforcing $\mathrm{O} \setminus R$. Constraint filtering (6) shows that removing those constraints leaves M_R stable for $A(\mathrm{P}, \mathrm{O})$, while the constraints themselves give $\mathrm{O} \setminus R \subseteq M_R$.

Suppose some $r \in R$ also belonged to M_R. Then M_R would satisfy the additional constraint `:- not` r. Constraint filtering (6) shows that adding this satisfied constraint preserves stability. But the resulting program is $\mathsf{enforce}(\mathrm{P}, (\mathrm{O} \setminus R) \cup \{r\}, \mathrm{O})$, so $R \setminus \{r\}$ would be a correction set, contradicting minimality of R. Therefore $M_R \cap \mathrm{O} = \mathrm{O} \setminus R$ and $\rho(M_R) = R$.

Weak constraints do not alter the stable models of $A(\mathrm{P}, \mathrm{O})$; they only rank them. Exactly those indexed by atoms in R are active in M_R, so its cost is $\sum_{r \in R} w_r = \mathcal{C}(R, \mathrm{W})$. □

We next recall the main-paper projection theorem.

Theorem 1 (Projection correspondence)**.** *Let* P *be a finite ground program,* O *a finite set of objective atoms, and* W *assign strictly positive release weights to all atoms in* O. *Then:*

($\Rightarrow$) *If* M *is an optimal stable model of* $\mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W})$*, then* $\rho(M)$ *is an optimal MCS (Def. 3), and the weak-constraint cost of* M *is* $\mathcal{C}(\rho(M), \mathrm{W})$.

($\Leftarrow$) *For every optimal MCS* R*, there exists an optimal stable model* M *of* $\mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W})$ *with* $\rho(M) = R$.

Distinct optimal stable models may have the same released-fact projection.

Proof. ($\Rightarrow$) Let M be an optimal stable model of $\mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W})$. Weak constraints do not alter stable models, so M is stable for $A(\mathrm{P}, \mathrm{O})$. Let $S = M \cap \mathrm{O} = \mathrm{O} \setminus \rho(M)$. The model M satisfies every constraint enforcing an atom in S; hence constraint filtering (6) makes M stable for $\mathsf{enforce}(\mathrm{P}, S, \mathrm{O})$. Therefore $\rho(M)$ is a correction set. Moreover, precisely the weak constraints indexed by $\rho(M)$ are active, so the model cost is $\mathcal{C}(\rho(M), \mathrm{W})$.

Suppose a correction set Q had lower cost. Since O is finite, Q contains an MCS $R \subseteq Q$. Positivity of the weights gives

$$\mathcal{C}(R, \mathrm{W}) \leq \mathcal{C}(Q, \mathrm{W}) < \mathcal{C}(\rho(M), \mathrm{W}).$$

Lemma S.3 supplies a stable model M_R whose weak-constraint cost is $\mathcal{C}(R, \mathrm{W})$, contradicting optimality of M. Thus $\rho(M)$ is a minimum-cost correction set and, by Lemma 1, an optimal MCS.

($\Leftarrow$) Let R be an optimal MCS. Lemma S.3 supplies a stable model M_R with $\rho(M_R) = R$ and cost $\mathcal{C}(R, \mathrm{W})$. If M_R were not optimal, finiteness of the ground program would let us choose a globally optimal stable model N of strictly lower cost. The already proved ($\Rightarrow$) direction now applies to this optimal N: $\rho(N)$ is an MCS and its cost equals the cost of N, so

$$\mathcal{C}(\rho(N), \mathrm{W}) < \mathcal{C}(R, \mathrm{W}),$$

contradicting optimality of R. Hence M_R is optimal and realises R exactly.

The two directions establish that ρ maps the optimal stable models onto exactly the optimal MCSes of Definition 3, preserving their costs. If $A(\mathrm{P}, \mathrm{O})$ is incoherent, constraint filtering implies that there are no correction sets, so both families are empty. Thus ρ is surjective onto the optimal MCSes in all cases. Injectivity is not guaranteed: for $\mathrm{P} = \{\texttt{a :- not b.}, \texttt{b :- not a.}\}$, $\mathrm{O} = \{\texttt{o}\}$, and any $w_\circ > 0$, the two optimal stable models $\{\texttt{a}, \texttt{o}\}$ and $\{\texttt{b}, \texttt{o}\}$ both project to the unique optimal MCS $\emptyset$. □

The optimal-repair result also yields a local justification for every released fact. We recall and prove the main-paper private-conflict proposition.

Proposition 1 (Conflict justification)**.** *Let* P *be a finite ground program and* O *a finite set of objective atoms. For every* $\mathrm{R} \in MCS(\mathrm{P}, \mathrm{O})$ *and every* $f \in \mathrm{R}$*, there is a MUS* $U_f \in MUS(\mathrm{P}, \mathrm{O})$ *such that* $U_f \cap \mathrm{R} = \{f\}$.

Proof. Recall the MUS–MCS hitting-set duality stated in the main paper: $MCS(\mathrm{P}, \mathrm{O})$ is exactly the family of inclusion-minimal hitting sets of $MUS(\mathrm{P}, \mathrm{O})$ (Alviano et al., 2023, Thm. 5.3). Let $R \in MCS(\mathrm{P}, \mathrm{O})$ and $f \in R$. Since $R \setminus \{f\}$ is not a hitting set, some $U_f \in MUS(\mathrm{P}, \mathrm{O})$ is disjoint from $R \setminus \{f\}$. Since R does hit every MUS, it follows that $f \in U_f$ and therefore $U_f \cap R = \{f\}$. This proves the private-conflict condition in the proposition. □

The certificate is local to the contested release: every other member of U_f was retained. It proves that adding f back without a compensating release restores a concrete conflict.

E OptABA-PC Guarantees

We recall the OptABA-PC guarantees theorem from the main paper.

Theorem 2 (OptABA-PC guarantees)**.** *Let* $\mathbf{V}$ *be finite,* O *a finite set of valid CI facts over* $\mathbf{V}$*, and* W *assign strictly positive release weights to every fact in* O*. Then* $\mathsf{opt}(\mathrm{P}_{\mathrm{CA}}, \mathrm{O}, \mathrm{W})$ *has an optimal stable model. For every such model* M*, put* $R = \rho(M)$ *and* $T = \mathrm{O} \setminus R$*. Then:*

(1) $\pi_G(M)$ *is a DAG on* $\mathbf{V}$ *satisfying every fact in* T*;*

(2) R *is an MCS with minimum release cost among all correction sets of* P_{CA} *w.r.t.* O*;*

(3) *for every* $f \in R$*, enforcing* $T \cup \{f\}$ *is incoherent, and a MUS* U_f *satisfies* $U_f \cap R = \{f\}$.

Conversely, every optimal MCS R *is* $\rho(M)$ *for some optimal stable model* M*, with the same cost.*

Proof. We first prove the existence of an optimal stable model. Proposition 3.3 of Russo et al. (2024) establishes a stable Causal ABA extension for every DAG on the fixed vertex set $\mathbf{V}$. Apply it to the edgeless DAG $G_\emptyset = (\mathbf{V}, \emptyset)$, which retains all vertices and has no edges. The stable ABA–ASP correspondence (Rapberger et al., 2024) supplies the corresponding stable choices for the inherited ASP core. With no objective facts selected, the finite positive or stratified reachability and Bayes-ball modules have their least closure, and no CI-enforcement constraint is activated. Hence $A(\mathrm{P}_{\mathrm{CA}}, \mathrm{O}) = \mathsf{enforce}(\mathrm{P}_{\mathrm{CA}}, \emptyset, \mathrm{O})$ has a stable model. Weak constraints preserve this nonempty feasible set and only rank its members. The program is finite ground, so its attainable costs form a nonempty finite set and have a minimum; therefore $\mathsf{opt}(\mathrm{P}_{\mathrm{CA}}, \mathrm{O}, \mathrm{W})$ has an optimal stable model.

Let M be any optimal stable model, let $R = \rho(M)$ and $T = \mathrm{O} \setminus R = M \cap \mathrm{O}$, and abbreviate $\mathcal{F}_{\mathrm{CA}} = CS(\mathrm{P}_{\mathrm{CA}}, \mathrm{O})$.

1. Graph fidelity. Weak constraints do not alter stable models, so M is stable for $A(\mathrm{P}_{\mathrm{CA}}, \mathrm{O})$. Because $T = M \cap \mathrm{O}$, M satisfies every constraint enforcing an atom in T. Constraint filtering (6) therefore gives $M \in AS(\mathsf{enforce}(\mathrm{P}_{\mathrm{CA}}, T, \mathrm{O}))$. Corollary 1 now yields $\pi_G(M) \in \mathrm{DAG}(\mathbf{V})$ and $\pi_G(M) \models T$, which is item 1.

2. Repair optimality. The ($\Rightarrow$) direction of Theorem 1 gives $R \in MCS(\mathrm{P}_{\mathrm{CA}}, \mathrm{O})$, identifies $\mathcal{C}(R, \mathrm{W})$ with the weak-constraint cost of M, and makes R minimum-cost among all MCSes. By definition, membership in MCS gives subset-minimality. Lemma 1 and the observation following

its proof show that the minimum cost over all correction sets equals the minimum cost over MCSes. Therefore

$$R \in MCS(\mathrm{P}_{\mathrm{CA}}, \mathrm{O}), \qquad \mathcal{C}(R, \mathrm{W}) = \min_{Q \in \mathcal{F}_{\mathrm{CA}}} \mathcal{C}(Q, \mathrm{W}),$$

This proves item 2.

3. Conflict justification. Fix $f \in R$. Since R is an MCS, $R \setminus \{f\}$ is not a correction set. Because

$$\mathrm{O} \setminus (R \setminus \{f\}) = T \cup \{f\},$$

$AS(\mathsf{enforce}(\mathrm{P}_{\mathrm{CA}}, T \cup \{f\}, \mathrm{O})) = \emptyset$, which is precisely the incoherence caused by reinstating f while retaining T. Proposition 1, proved above by MUS–MCS hitting-set duality, supplies a MUS U_f with $U_f \cap R = \{f\}$. Hence $f \in U_f$, while every other member of U_f lies in $T = \mathrm{O} \setminus R$. Thus U_f is a conflict contained in the reinstated set $T \cup \{f\}$ and certifies that incoherence, proving item 3.

Finally, let R' be any optimal MCS. The ($\Leftarrow$) direction of Theorem 1 supplies an optimal stable model N of $\mathsf{opt}(\mathrm{P}_{\mathrm{CA}}, \mathrm{O}, \mathrm{W})$ such that $\rho(N) = R'$ and whose weak-constraint cost is $\mathcal{C}(R', \mathrm{W})$. This proves the converse clause of Theorem 2. The witness need not be unique, as stated in Theorem 1. □

F Contestability and Redress

Let $\mathcal{F} = CS(\mathrm{P}, \mathrm{O})$, and partition it into

$$K_f = \{R \in \mathcal{F} : f \notin R\}, \qquad L_f = \{R \in \mathcal{F} : f \in R\}.$$

Because a repair lists released facts, K_f contains the repairs retaining f, whereas L_f contains those releasing it. Throughout this section, O is finite, all release weights are positive, and $f \in \mathrm{O}$. For hard retention, write

$$A_f = \mathsf{opt}_f(\mathrm{P}, \mathrm{O}, \mathrm{W}) = \mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W}) \cup \{\texttt{:- not } f\}.$$

We recall the two contestability properties and their OptABA-PC guarantee from the main paper.

Property 1 (Release-cost contestability)**.** *For a weighted repair method over $\mathcal{F}$, let $\mathrm{Rep}_{\mathrm{W}} \subseteq \mathcal{F}$ denote the repairs returned under weights* W*. Suppose* W′ *differs from* W *only in that $w'_f > w_f$. The method is* release-cost contestable at f *if its returned repairs can change only towards retaining f:*

$$\{R \in \mathrm{Rep}_{\mathrm{W}} : f \notin R\} \subseteq \mathrm{Rep}_{\mathrm{W}'}, \qquad \{R \in \mathrm{Rep}_{\mathrm{W}'} : f \in R\} \subseteq \mathrm{Rep}_{\mathrm{W}}\,.$$

Property 2 (Fact-enforcement contestability)**.** *An ASP repair method based on* $\mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W})$ *is* fact-enforcement contestable at f *if*

$$\{\rho(M) \mid M \textit{ is optimal for } \mathsf{opt}_f(\mathrm{P}, \mathrm{O}, \mathrm{W})\} = \arg\min_{R \in \mathcal{F},\, f \notin R} \mathcal{C}(R, \mathrm{W}).$$

It is total *on a fact domain if, for every fact f in that domain, some $R \in \mathcal{F}$ satisfies $f \notin R$. Thus making f mandatory returns exactly the best repairs that retain it.*

Proposition 2 (OptABA-PC contestability)**.** *Let* $\mathbf{V}$ *and* O *be finite, let every member of* O *be a valid CI fact over* $\mathbf{V}$*, put* $\mathrm{P}_{\mathrm{CA}} = \mathrm{P}_{\mathrm{CA}}(\mathbf{V}, \mathrm{O})$ *as in Equation* (1)*, and let* $\mathrm{W} = \{(o, w_o) \mid o \in \mathrm{O},\ w_o > 0\}$*. Then OptABA-PC satisfies Properties 1 and 2 for every $f \in \mathrm{O}$. Moreover, fact-enforcement contestability is total:* $\mathsf{opt}_f(\mathrm{P}_{\mathrm{CA}}, \mathrm{O}, \mathrm{W})$ *has an optimal stable model for every $f \in \mathrm{O}$.*

Proof. Fix the finite sets $\mathbf{V}$ and O of the OptABA-PC instance, where every member of O is a valid CI fact over $\mathbf{V}$, let $\mathrm{P} = \mathrm{P}_{\mathrm{CA}}(\mathbf{V}, \mathrm{O})$ be the unrestricted-DAG program in Equation (1), let W assign one strictly positive release weight to every objective atom, and fix $f \in \mathrm{O}$.

Release-cost contestability. For either positive weight assignment $\overline{\mathrm{W}} \in \{\mathrm{W}, \mathrm{W}'\}$, both directions of Theorem 1 identify the returned projections with the minimum-cost MCSes, and Lemma 1 identifies those with all minimum-cost correction sets. Consequently,

$$\mathrm{Rep}_{\overline{\mathrm{W}}} = \arg\min_{R \in \mathcal{F}} \mathcal{C}(R, \overline{\mathrm{W}}).$$

Let $\delta = w'_f - w_f > 0$, and abbreviate $\mathcal{C}(R) = \mathcal{C}(R, \mathrm{W})$ and $\mathcal{C}'(R) = \mathcal{C}(R, \mathrm{W}')$. Because W′ changes only w_f, $\mathcal{C}'(R) = \mathcal{C}(R)$ for $R \in K_f$, while $\mathcal{C}'(R) = \mathcal{C}(R) + \delta$ for $R \in L_f$. If either K_f or

L_f is empty, every repair receives the same cost change: no repair changes cost when $L_f = \emptyset$, and every repair gains δ when $K_f = \emptyset$. In either case $\mathrm{Rep}_{\mathrm{W}'} = \mathrm{Rep}_{\mathrm{W}}$, so both required inclusions are immediate. Assume henceforth that both classes are nonempty. Since $\mathcal{F}$ is finite, define the attained within-class minima and their complete sets of minimisers by

$$k = \min_{R \in K_f} \mathcal{C}(R), \quad K_f^* = \arg\min_{R \in K_f} \mathcal{C}(R),$$
$$\ell = \min_{R \in L_f} \mathcal{C}(R), \quad L_f^* = \arg\min_{R \in L_f} \mathcal{C}(R).$$

The ranking inside K_f is unchanged because none of its costs changes, and the ranking inside L_f is unchanged because the same δ is added to every one of its costs. Hence K_f^* and L_f^* remain the respective within-class minimisers, while the best class costs change from k and ℓ under W to k and $\ell + \delta$ under W′. The following five cases exhaust the possible comparisons.

1. If $k < \ell$, then $\mathrm{Rep}_{\mathrm{W}} = K_f^*$. Since $k < \ell < \ell + \delta$, also $\mathrm{Rep}_{\mathrm{W}'} = K_f^*$.
2. If $k = \ell$, then $\mathrm{Rep}_{\mathrm{W}} = K_f^* \cup L_f^*$. Since $k < \ell + \delta$, the increase breaks the tie in favour of retention and $\mathrm{Rep}_{\mathrm{W}'} = K_f^*$.
3. If $\ell < k$ and $\delta < k - \ell$, then $\ell + \delta < k$, so $\mathrm{Rep}_{\mathrm{W}} = \mathrm{Rep}_{\mathrm{W}'} = L_f^*$.
4. If $\ell < k$ and $\delta = k - \ell$, then $\ell + \delta = k$. Therefore $\mathrm{Rep}_{\mathrm{W}} = L_f^*$, while $\mathrm{Rep}_{\mathrm{W}'} = K_f^* \cup L_f^*$: the best retaining repairs join the old releasing optima at a tie.
5. If $\ell < k$ and $\delta > k - \ell$, then $k < \ell + \delta$. Therefore $\mathrm{Rep}_{\mathrm{W}} = L_f^*$ and $\mathrm{Rep}_{\mathrm{W}'} = K_f^*$.

Cases 1 and 2 are exactly the cases in which an old optimum retains f: every such repair belongs to K_f^* and remains a new optimum. In Cases 3–5 there is no old retaining optimum. Since $K_f = \{R \in \mathcal{F} : f \notin R\}$, this establishes the first inclusion in Property 1:

$$\{R \in \mathrm{Rep}_{\mathrm{W}} : f \notin R\} = \mathrm{Rep}_{\mathrm{W}} \cap K_f \subseteq \mathrm{Rep}_{\mathrm{W}'}\,.$$

Cases 3 and 4 are exactly the cases in which a new optimum releases f: every such repair belongs to L_f^* and was already an old optimum. In Cases 1, 2, and 5 there is no new releasing optimum. Since $L_f = \{R \in \mathcal{F} : f \in R\}$, this establishes the second inclusion:

$$\{R \in \mathrm{Rep}_{\mathrm{W}'} : f \in R\} = \mathrm{Rep}_{\mathrm{W}'} \cap L_f \subseteq \mathrm{Rep}_{\mathrm{W}}\,.$$

Together, the two inclusions prove release-cost contestability at f.

Fact-enforcement contestability. By constraint filtering (6), adding `:- not f` removes precisely the stable models with $f \notin M$, equivalently those with $f \in \rho(M)$, and changes no weak-constraint cost. Thus every surviving projection lies in K_f.

($\subseteq$) Let M be an optimal stable model of A_f, let $R = \rho(M)$, and let $S = M \cap \mathrm{O}$. Constraint filtering (6) shows that removing the hard constraint leaves M stable for $A(\mathrm{P}, \mathrm{O})$. Applying the same identity when adding the constraints enforcing S shows that R is a correction set; since $f \in M$, $R \in K_f$. Suppose some $Q \in K_f$ were cheaper. Because O is finite, Q contains an MCS $R' \subseteq Q$, which also belongs to K_f. Lemma S.3 supplies a stable model $M_{R'}$ with projection R' and cost at most $\mathcal{C}(Q, \mathrm{W})$. Because $f \notin R'$, the model satisfies the new hard constraint, and constraint filtering preserves its stability when that constraint is added. This contradicts optimality of M. Hence $R \in \arg\min_{Q \in K_f} \mathcal{C}(Q, \mathrm{W})$.

($\supseteq$) Let $R \in \arg\min_{Q \in K_f} \mathcal{C}(Q, \mathrm{W})$. A strict correction subset $R' \subset R$ would also lie in K_f and, by positive weights, would be cheaper; hence R is an MCS. Lemma S.3 supplies a stable model M_R with projection R and cost $\mathcal{C}(R, \mathrm{W})$. Since $f \notin R$, M_R satisfies the hard constraint, and constraint filtering preserves stability after it is added. If a lower-cost stable model N of A_f existed, removing the hard constraint and then adding the constraints enforcing $N \cap \mathrm{O}$, using the two corresponding applications of (6), would show that $\rho(N) \in K_f$ is a correction set. Its cost equals the lower model cost, contradicting the choice of R. Thus M_R is optimal for A_f, proving the reverse inclusion.

Totality. It remains to prove that K_f is nonempty for every valid CI fact f. Corollary 1 shows that P_{CA} admits every DAG on the fixed vertex set $\mathbf{V}$ before CI enforcement. If $f = (X \perp\!\!\!\perp Y \mid \mathbf{Z})$, the

edgeless (fully disconnected) DAG $(\mathbf{V}, \emptyset)$ satisfies f. If $f = (X \not\perp\!\!\!\perp Y \mid \mathbf{Z})$, the DAG $(\mathbf{V}, \{(X, Y)\})$, in which every other vertex is isolated, satisfies f because its direct $X \to Y$ path has no internal node and validity gives $X, Y \notin \mathbf{Z}$. In either case, Corollary 1 with $T = \{f\}$ supplies a stable model enforcing f; releasing every other fact therefore gives a correction set in K_f. Thus the hard-retention program is coherent. It is finite ground, so its nonempty finite set of attainable costs has a minimum and it has an optimal stable model, proving totality. □

A fact-enforcement contestation gives categorical redress by making f mandatory. A user may instead make a graded contestation, increasing the cost of releasing f while leaving it possible. The operational question is then how much that cost must increase before retaining f becomes at least as good as releasing it. We first define the relevant quantity generally and then prove that it is the exact switching threshold for OptABA-PC.

We recall the contestability-margin definition from the main paper.

Definition 4 (Contestability margin)**.** *Let $\mathcal{F}$ be a finite family of repairs and* W *assign a finite release cost to every objective atom. For any fact f for which $\mathcal{F}$ contains both a repair retaining f and a repair releasing f, define*

$$c_{\mathrm{keep}}(f) = \min_{R \in \mathcal{F},\, f \notin R} \mathcal{C}(R, \mathrm{W}), \qquad c_{\mathrm{rel}}(f) = \min_{R \in \mathcal{F},\, f \in R} \mathcal{C}(R, \mathrm{W}).$$

If at least one globally optimal repair under W *releases f, then $c_{\mathrm{rel}}(f) \le c_{\mathrm{keep}}(f)$. The* contestability margin *of f is $\mu(f) = c_{\mathrm{keep}}(f) - c_{\mathrm{rel}}(f) \ge 0$.*

We now specialise to $\mathrm{P} = \mathrm{P}_{\mathrm{CA}}$. Proposition 2 makes K_f nonempty for every valid CI fact, and the all-release correction set belongs to L_f. Since O is finite and its release costs are finite, $\mathcal{F}$ is finite and every repair has finite cost; both category minima are therefore attained, as required by the definition. Proposition S.1 below shows exactly how their difference governs OptABA-PC's response.

Proposition S.1 (OptABA-PC contestation threshold)**.** *Let f be a valid CI fact released by at least one optimal OptABA-PC repair under* W*. For $\delta \ge 0$, let W_δ replace w_f by $w_f + \delta$, leaving every other release cost unchanged, and let Rep_δ be the repairs returned after re-solving OptABA-PC under W_δ. Then:*

1. *if $\delta < \mu(f)$, $\mathrm{Rep}_\delta \subseteq L_f$: every returned repair still releases f;*
2. *if $\delta = \mu(f)$, Rep_δ contains both a retaining and a releasing repair;*
3. *if $\delta > \mu(f)$, $\mathrm{Rep}_\delta \subseteq K_f$: every returned repair retains f.*

With integer implementation weights, $\mu(f) + 1$ is therefore the smallest integer increase that forces every returned repair to retain f.

Proof. Raising only w_f leaves the cost of every retaining repair unchanged and adds δ to every releasing repair. Thus their respective best costs under W_δ are

$$c_{\mathrm{keep}}(f) \quad \text{and} \quad c_{\mathrm{rel}}(f) + \delta.$$

Their difference is $c_{\mathrm{keep}}(f) - [c_{\mathrm{rel}}(f) + \delta] = \mu(f) - \delta$. Lemma 1 and Theorem 1 identify the globally minimum-cost correction sets under W_δ with Rep_δ. Both K_f and L_f are finite and nonempty, so both displayed category minima are attained.

1. Below the margin. If $\delta < \mu(f)$, then $c_{\mathrm{rel}}(f) + \delta < c_{\mathrm{keep}}(f)$. Hence no retaining repair is globally optimal and $\mathrm{Rep}_\delta \subseteq L_f$.

2. At the margin. If $\delta = \mu(f)$, then $c_{\mathrm{rel}}(f) + \delta = c_{\mathrm{keep}}(f)$. A minimiser in each of L_f and K_f is therefore globally optimal, so Rep_δ contains both a releasing and a retaining repair.

3. Above the margin. If $\delta > \mu(f)$, then $c_{\mathrm{keep}}(f) < c_{\mathrm{rel}}(f) + \delta$. Hence no releasing repair is globally optimal and $\mathrm{Rep}_\delta \subseteq K_f$.

Integer threshold. If all weights are integers, both category costs and $\mu(f)$ are integers. The smallest integer increase strictly greater than $\mu(f)$ is therefore $\mu(f) + 1$, which by item 3 is the smallest integer increase forcing every returned repair to retain f. □

The margin is therefore not a graph-quality measure. It is an actionable threshold: $\mu(f) = 0$ exposes an already optimal retaining alternative, $\mu(f)$ is the increase that creates a retaining–releasing tie, and any larger increase forces retention.

Beyond enabling an individual contestation, the margin supports a comparative reading of inconsistency. Logical incoherence remains binary, but $\mu(f)$ quantifies its weighted consequence for a released fact: it is the additional repair cost incurred by resolving the conflict in favour of retaining f, rather than releasing it. Thus, within a fixed trace and weighting scheme, larger margins identify releases that are more strongly supported by the competing evidence and more resistant to reinstatement. The normalised margin $\mu(f)/w_f$ gives the proportional increase in f's release cost required to reach a retaining tie, allowing comparisons across facts and traces that use the same weight semantics. It should therefore be interpreted as relative weighted tension within the repair objective, not as a probability that f is false or as an absolute degree of logical inconsistency. Appendix G.3 evaluates both hard-retention responses and these margins (Tables 6 and 8).

These guarantees concern the enforced CI commitments and have a precise consequence for their graph witnesses. Fix an optimal repair R releasing f and put $T = \mathrm{O} \setminus R$. By Theorem 2(3), enforcing $T \cup \{f\}$ is incoherent; Corollary 1 therefore implies that no DAG satisfying T can also satisfy f. Hard retention, or a cost increase $\delta > \mu(f)$, returns only graphs satisfying f, so it excludes every graph compatible with that releasing repair. A graph belonging to a different, already optimal repair retaining f may remain when $\mu(f) = 0$. Below the margin, releasing optima remain optimal; at the margin, retaining and releasing optima coexist. Thus changing the enforced facts can force a change of graph witness, without guaranteeing improved graph accuracy or recovery of the generating DAG.

G Additional Empirical Evidence

This section complements the primary experiments and formal results. It begins by instantiating the repair, private-conflict, and contestability constructions on the four-node example from the main paper. It then gives the full experimental protocol and metric definitions, summarises the fact enforcement comparison across methods and datasets, compares optimal and heuristic ABA repair, reports fact-level redress, distinguishes the repair claim from MPC's procedural use of tests and ASPCR-DAG's different objective, and concludes with the full reconstruction, fact-quality, and computational results.

G.1 Four-Node Worked Example

The main paper reports 14 CI facts and their strengths for the variables (E, R, O, I). We multiply those reported strengths by 1000 for illustrative integer costs; the empirical weight scaling is given in Table 3. Exhaustive enumeration of all 543 labelled DAGs on these variables gives the unique OptABA-PC baseline repair

$$R = \{\texttt{indep(e,o,\{i\}), indep(r,i,\{e\}), indep(o,i,\{\})}\}$$

of cost 2040. Its unique graph witness has edges $E \to O$, $R \to O$, and $O \to I$. Table 2 gives one private MUS and the hard-retention cost and margin for every released fact. Each listed MUS intersects R only in the contested fact.

The corresponding minimum-cost hard-retention repairs are, in table order,

$R_1 = \{\texttt{indep(e,i,\{o\}), indep(r,i,\{e\}), indep(o,i,\{\}), indep(e,r,\{\})}\}$,
$R_2 = \{\texttt{indep(e,o,\{i\}), indep(e,i,\{o\}), indep(o,i,\{\}), indep(e,r,\{\})}\}$,
$R_3 = \{\texttt{dep(e,o,\{\}), dep(e,o,\{r\}), dep(r,i,\{\}), indep(e,i,\{o\})}\}$.

Each differs from R and retains the contested fact. The private MUS proves that reinstatement alone is incoherent. Increasing the contested fact's release weight by 30, 500, or 870, respectively, makes the baseline and hard-retention costs tie; increments of 31, 501, or 871 force every optimum to retain it. A separate 13-fact variant, obtained by omitting `indep(e,r,{})` from the trace, is also exhaustively checked. It has 18 MCSes, a unique optimal released-fact projection of cost 1570, and four compatible DAGs. Making the released fact `indep(e,i,{o})` mandatory raises the proved optimum to 2040 and yields a disjoint set of 13 compatible DAGs. This fact is false for the

Table 2: Contestation certificates and exact integer margins for the three facts released in the 14-fact four-node trace. The weights are the main-paper strengths multiplied by 1000; c_{keep} is the minimum repair cost after making the contested fact mandatory, and $\mu(f) = c_{\text{keep}}(f) - 2040$.

Contested fact f	w_f	c_{keep}	$\mu(f)$	One private MUS U_f
`indep(e,o,{i})`	1000	2070	30	{f, `dep(e,o,{})`, `indep(e,i,{o})`}
`indep(r,i,{e})`	530	2540	500	{`dep(r,i,{})`, f, `indep(e,r,{})`}
`indep(o,i,{})`	510	2910	870	{`indep(r,i,{o})`, `dep(r,i,{})`, f}

generating DAG in the main paper, whose direct edge $E \to I$ remains active given O; neither graph family contains that DAG. The example demonstrates enforced commitment, not truth restoration.

G.2 Experimental Protocol and Metrics

Benchmark choice. Using the benchmark design of Causal ABA (Russo et al., 2024), the fixed-network study includes Cancer (5 nodes, 4 edges), Earthquake $(5, 4)$, Survey $(6, 6)$, and Asia $(8, 8)$ from the bnlearn repository (Scutari, 2014). These small, standard Bayesian networks include medical, legal, and policy decision models derived from published studies or expert elicitation. They are appropriate for this evaluation for three reasons: their known DAGs and conditional probability tables (CPTs) provide ground truth for both CI facts and graph reconstruction; repeated finite samples expose the test errors and conflicts that motivate repair; and reusing the benchmarks and sampling design of (Russo et al., 2024) makes the comparison with ABA-PC a controlled extension rather than a change of empirical regime. We add Erdős–Rényi (ER) (Erdős & Rényi, 1959) and scale-free (SF) (Barabási & Albert, 1999) DAGs with five and eight nodes to vary topology beyond four fixed networks and to evaluate the MPC (Colombo & Maathuis, 2014) and FGS (Ramsey et al., 2017) reference methods under the same generated samples.

Data generation. For each dataset we use 50 seeds, 2026–2075, and generate $N = 5000$ independent observations per seed. For a bnlearn benchmark, we load its BIF file, including the published DAG and CPTs, and forward-sample a new data matrix for each seed using the sampling protocol of (Russo et al., 2024). Thus the BN is fixed and only the finite sample changes. For a primary synthetic run, the seed determines both a new graph and its data-generating distribution. An ER graph is formed with n edges and given a random acyclic orientation and node permutation; an SF graph is generated by directed Barabási–Albert preferential attachment with one attachment per added node and then randomly permuted. Each node is categorical with at least two states; its allowed cardinality is reduced when it participates in a larger parent configuration, using 20 as the implementation's CPT-enumeration target. For each node and parent configuration, its probability vector is drawn from a symmetric Dirichlet distribution whose concentration is sampled uniformly from $[1, 5]$, after which the BN is forward-sampled. Categorical states are numerically encoded and columns are aligned with the generating adjacency matrix. Before CI testing, the bnlearn codes are additionally standardised; this is a one-to-one relabelling and therefore leaves the contingency counts for the G^2 CI test (Spirtes et al., 2000) unchanged. FGS independently factorises every categorical column into integer levels for its BDeu score.

Methods and matched evidence. Our proposed OptABA-PC, ABA-PC (Russo et al., 2024), and MPC (Colombo & Maathuis, 2014) use the likelihood-ratio G^2 conditional-independence test (Wilks, 1935; Spirtes et al., 2000) at $\alpha = 0.01$. Within a seed, ABA-PC and OptABA-PC receive exactly the same recorded CI facts and p-value-derived release strengths (as in (Russo et al., 2024)), as well as the same time limits; their paired comparison therefore changes only the repair rule. MPC uses the same sample and test configuration but consumes test outcomes procedurally. FGS (Ramsey et al., 2017) uses Tetrad's native discrete BDeu score on the same categorical sample and returns one score-selected CPDAG; because it exposes no CI trace, it is a graph-reconstruction rather than fact-repair baseline. ASPCR-DAG (Hyttinen et al., 2014) uses its native Bayesian log-weighted CI configuration (prior independence 0.4 and Bayesian α parameter 20) on the same saved

Table 3: Baseline descriptions and final method settings. Shared data, seed, CI, and computational settings are given in the surrounding protocol.

Method	Role and implementation	Final settings
MPC	Majority-PC (Colombo & Maathuis, 2014), an order-independent PC variant, learns a skeleton by CI testing, orients colliders by majority evidence across separating sets, and applies Meek propagation to return one CPDAG.	Stable skeleton search; G^2 test; $\alpha = 0.01$; majority collider rule (implementation `uc_rule=5`); collider priority 2; no conditioning-depth limit.
ABA-PC	Causal ABA-PC (Russo et al., 2024) receives MPC's CI trace and releases facts from lowest to highest strength until its Bayes-ball Causal ABA program is coherent.	Same MPC trace; no conditioning-set-size multiplier; strength $s(p) = 1 - p/(2\alpha)$ for $p < \alpha$ and $s(p) = (\alpha - p/2 - 1/2)/(\alpha - 1)$ otherwise; integer weight $\max\{1, \mathrm{round}(9{,}999{,}999s)\}$; incremental Bayes-ball encoding.
OptABA-PC	Our method uses the same facts, weights, and candidate-DAG encoding as ABA-PC, replacing sequential release with one minimum-cost correction-set optimisation.	MUS-reified weak constraints; *clingo* branch-and-bound (`bb`); `optN` enumeration of optimal stable models; minimum total released weight; four solver threads.
FGS	Fast Greedy Search (Ramsey et al., 2017) is the Tetrad/`py-causal` fast implementation of GES (Chickering, 2002b); it greedily inserts and removes edges under a score and returns one CPDAG.	BDeu score (`bdeu-score`); discrete-data interface; sample prior 15; structure prior 1; unrestricted maximum degree (`-1`); faithfulness assumed; symmetric first step disabled.
ASPCR-DAG	The ASP method of Hyttinen et al. (2014), restricted here to causally sufficient DAGs, assigns a cost to every graph–test disagreement and returns one optimal DAG.	Exhaustive conditioning schedule $n - 2$; Bayesian CI test; log weights; prior independence 0.4; Bayesian $\alpha = 20$; acyclic sufficient encoding; *clingo* `crafty` configuration; comparison time limit 300 s.

samples. Its graph comparison and forced-fact sensitivity audit cover Cancer, Earthquake, Survey, ER(5), and SF(5), with 50 DAG-restricted (otherwise it could return partial ancestral graphs under no causal sufficiency assumption, allowing for latent confounders) runs per dataset.

Hyperparameters. The configuration of parameters in Table 3 was fixed from the standard published implementation or the Causal ABA protocol. In particular, G^2 with $\alpha = 0.01$ is the Causal ABA setting used by MPC, ABA-PC, and OptABA-PC; MPC uses the standard stable-majority rule; FGS uses Tetrad's native discrete BDeu configuration listed in the table; and ASPCR-DAG uses the authors' Bayesian log-weighted configuration.

Computation. All ABA repair runs use our proposed Bayes-ball encoding, a 300-second solve limit, a separate 120-second compatible-set evaluation limit, and four *clingo* threads. ASPCR-DAG is omitted above six nodes because preliminary runs exceeded one hour, well beyond the time limit in Table 3. This is also reported in the original paper (Hyttinen et al., 2014). At eight nodes, the $\binom{n}{2}2^{n-2} = 1792$ possible CI constraints define a conceptual space of 2^{1792} fact subsets, motivating both the limits and the smaller scope of the exact contestability audit. Tables 5 and 7 audit all 400 (50 seeds $\times$ 8 datasets) primary G^2 traces. The exact decision audit in Table 6 uses 250 proved ABA instances and makes each released OptABA-PC fact mandatory in turn. The separate ASPCR-DAG sensitivity audit (Table 6) re-solves all 4,330 failed facts in 250 proved native outputs. MPC has no consistent DAG extension for 12 endpoint-invalid Survey outputs and one endpoint-invalid ER(8) output, so only the affected DAG metrics display paired sample sizes of 38 and 49.

Table 4 isolates the efficiency differences between the original active-path (Russo et al., 2024) and our proposed Bayes-ball encodings under the matched primary configuration. Across 50 seeds on each of the eight datasets, both implementations receive identical test traces and weights and use the same sequential ABA-PC repair policy; every pair that completes returns the same retained-fact count and weight. Bayes-ball completes all 400 runs, while active-path materialisation times out on three ER(8) and two SF(8) runs. Its mean end-to-end time is significantly lower on Cancer, Survey, Asia, ER(8), and SF(8), indistinguishable on ER(5), and higher on the very small Earthquake and

Table 4: Matched encoding-efficiency ablation over seeds 2026–2075, with $N = 5000$, G^2 tests at $\alpha = 0.01$, identical test traces and weights, and the same sequential ABA-PC repair policy. Only the causal-consistency encoding changes. Times are mean $\pm$ standard deviation for completed seed pairs and include program construction and solving; "BB/original" is the ratio of those means, so values below one favour Bayes-ball. Completion is over all 50 runs. Bold marks the significantly faster implementation under paired two-sided Wilcoxon tests with BH adjustment across datasets (*, **, and *** for adjusted $p < 0.05$, 0.01, and 0.001). Every completed pair returned identical retained-fact counts, weights, and model counts.

Dataset	Completed		Paired n	End-to-end time (s)		BB/original
	Active paths	Bayes-ball		Active paths	Bayes-ball	
Cancer (5)	50/50	50/50	50	0.118 ± 0.027	$\mathbf{0.102 \pm 0.014}^{***}$	$0.86\times$
Earthquake (5)	50/50	50/50	50	$\mathbf{0.068 \pm 0.013}^{***}$	0.099 ± 0.013	$1.45\times$
Survey (6)	50/50	50/50	50	0.385 ± 0.119	$\mathbf{0.148 \pm 0.058}^{***}$	$0.38\times$
Asia (8)	50/50	50/50	50	51.66 ± 50.27	$\mathbf{2.74 \pm 4.48}^{***}$	$0.05\times$
ER (5)	50/50	50/50	50	0.078 ± 0.029	0.084 ± 0.007	$1.08\times$
ER (8)	47/50	50/50	47	39.72 ± 58.73	$\mathbf{1.26 \pm 4.14}^{***}$	$0.03\times$
SF (5)	50/50	50/50	50	$\mathbf{0.075 \pm 0.026}^{***}$	0.099 ± 0.096	$1.32\times$
SF (8)	48/50	50/50	48	34.43 ± 61.81	$\mathbf{0.867 \pm 2.488}^{***}$	$0.03\times$

SF(5) instances. The largest reductions occur at eight nodes: 94.7% on Asia and at least 96.8% on the paired synthetic runs. Thus the benefit is a marked improvement in scaling rather than a uniform speedup on negligible-runtime instances. The formal proof and independent exhaustive checker establish that the Bayes-ball rules enforce the same d-separation semantics without materialising active paths.

Fact and graph metrics. A recorded CI fact is correct when its independence or dependence relation agrees with d-separation in the generating DAG. Fact precision treats retained correct and false facts as true and false positives; fact recall additionally counts released correct facts as false negatives; FactF1 is their harmonic mean. Graph F1 is the harmonic mean of edge precision and recall, and Structural Hamming Distance (SHD) (Tsamardinos et al., 2006) counts adjacency and orientation edits relative to the generating graph. Because observational data identify a Markov-equivalence class, we score CPDAGs and, where the method returns a CPDAG, its *consistent DAG extensions*: DAGs in the represented class that preserve its adjacencies and compelled arrowheads while orienting reversible edges. ABA-PC and OptABA-PC are scored over the compatible DAGs saved within the evaluation budget; ASPCR-DAG is scored on its single returned DAG. Average, best, and worst values describe variation within the saved structural set, with the partial-enumeration qualification stated below.

MPC and FGS natively return a CPDAG rather than one oriented DAG. The facts selected by ABA-PC and OptABA-PC can likewise admit several DAGs, possibly spanning several CPDAGs. Table 11 therefore compares equivalence-class reconstruction and then the relevant DAG outputs: every compatible DAG for ABA-PC/OptABA-PC, every consistent extension of the CPDAG returned by MPC/FGS, and the single DAG returned by ASPCR-DAG. Tables 14–17 report SHD and F1 variation across the corresponding CPDAGs and DAGs. All DAGs in one CPDAG entail the same CI model, so compatibility is class-level; their edge orientations can nevertheless have different SHD and F1 against the generating DAG.

Using PC-based methods, returned structures need not satisfy the CI evidence used during discovery. For ABA-PC and OptABA-PC, the reported compatible CPDAGs and DAGs satisfy every retained post-repair fact. For MPC and ASPCR-DAG, compatibility requires satisfaction of every test in the complete recorded native trace. It is undefined for score-based FGS and is shown as "–". Full-trace enforcement permits no released facts, whereas selected-set enforcement tests only the commitments exposed by a repair. We report the seed-level mean $\pm$ standard deviation.

Statistical analysis. For every reported method comparison, we align runs by dataset and seed, discard a pair only when that outcome is unavailable for either method, and apply a two-sided Wilcoxon signed-rank test (Wilcoxon, 1945) to the paired values. Benjamini–Hochberg false-discovery-rate adjustment (Benjamini & Hochberg, 1995) is performed within each dataset–metric

Table 5: Evidence correspondence, run completion, and repair coverage. A trace is feasible if some DAG satisfies every recorded CI outcome; a full-trace output is the method's returned graph or class satisfying all of them. A timeout incumbent is a valid saved repair whose optimality was not proved before the limit. An explicit repair names excluded facts, and selected-repair enforcement verifies every retained fact against the output. ASPCR-DAG uses its primary matched run set; FGS has no CI trace. A dash denotes an undefined object.

Method	Run status	Feasible traces	Full-trace matches	Explicit repairs	Selected repair enforced
OptABA-PC	400/400 saved: 368 completed, 32 timeout incumbents	88/400	88/400	312/312	400/400
ABA-PC	400/400 saved and completed	88/400	88/400	312/312	400/400
ASPCR-DAG	250/250 saved; all optima proved	32/250	32/250	–	–
MPC	400/400 saved and completed	88/400	80/400	–	–
FGS	400/400 saved and completed	–	–	–	–

family across the available comparisons; adjusted $p < 0.05$ is treated as significant. An all-zero paired difference is assigned $p = 1$. Stars compare OptABA-PC with ABA-PC and daggers compare other methods with OptABA-PC. The only reduced paired samples are the MPC DAG-extension metrics identified above and locally in their tables. Table 1 reports within-seed differences and relative effect sizes; Tables 9 and 10 supply their absolute scale and inferential annotations.

Execution environment. Experiments were run CPU-only on Ubuntu 22.04.5 LTS (Linux 6.8.0-124, x86-64) on one Intel Xeon w5-2455X socket with 12 physical cores, 24 hardware threads, a 3.20 GHz base frequency and 4.60 GHz maximum frequency, and 128 GB RAM (125 GiB visible to the operating system); no GPU was used. The primary environment used Python 3.12.1, *clingo* 5.8.0, NumPy 1.26.4, pandas 2.2.3, SciPy 1.15.3, scikit-learn 1.6.1, NetworkX 3.4.2, `causal-learn` 0.1.4.1, pgmpy 1.0.0, `py-causal` 1.2.1, and OpenJDK 19.0.2. ASPCR-DAG used R 4.1.2 with `pcalg` 2.7-11, `graph` 1.72.0, and `Rgraphviz` 2.38.0.

G.3 Fact Enforcement Comparison

Table 5 reports run completion, input feasibility, output correspondence, and repair coverage. Table 6 distinguishes the decision evidence each method supplies. Table 7 then places the 400-trace OptABA-PC–ABA-PC cost comparison beside MPC's output enforcement statistics, while Table 8 gives the dataset-level forced-fact margins.

Table 5 states the two relevant counts directly. A *feasible trace* is an input for which at least one DAG satisfies every recorded outcome; a *full-trace output* is the returned graph or Markov class actually doing so. Only 88/400 primary G^2 traces are feasible. ABA-PC and OptABA-PC match all 88 whenever an output is present because both expose and enforce a selected repair. MPC returns a graph on every input but matches only 80 traces, leaving eight avoidable failures among the 88 feasible cases. In the primary ASPCR-DAG setting, the proved soft-objective optimum has zero disagreement on 32/250 traces; positive optimum cost establishes that the remainder are infeasible under its DAG encoding. FGS has no CI trace to audit.

Across all 400 runs, 368 completed and 32 stopped at the optimisation limit after saving a valid incumbent: two on Asia, 19 on ER(8), and 11 on SF(8). Compatible-set enumeration is a separate post-solve step: the stored primary ABA/OptABA-PC metadata flags every such enumeration as budget-limited, including all 50 SF(8) instances. The saved set sizes and set-average graph metrics are consequently partial-enumeration summaries; this does not affect the selected facts, saved repair consistency, or the separately proved exact audits below.

Table 6: Decision and forced-fact evidence. OptABA-PC, ABA-PC, and MPC use the exact 250-trace G^2 audit; ASPCR-DAG uses a separate 250-run native Bayesian audit. A forced OptABA-PC solve is a user contestation and $\mu(f)/w_f$ is its exact threshold. ASPCR-DAG's forced solve is a post-hoc sensitivity test of a soft graph-disagreement objective, so its gap/w_f is not numerically comparable. ABA-PC and MPC define neither category optima nor a forced-fact certificate. Brackets give $[Q_1, Q_3]$; a dash means that the object is undefined. Bold marks OptABA-PC's exact certified result.

Method	Scope	Baseline decision evidence	Forced-fact evidence	Certificate supplied
OptABA-PC	G^2, 250	Minimum repair on **111/111** inconsistent traces; lower cost than ABA-PC on **108**, tied on 3.	**742/742** proved; zero gaps 0/742; median $\mu/w_f = 0.455$ [0.182,0.912]. 11.6% are ≤ 0.1; 22.5% are > 1.	**Exact user-contestation threshold.**
ABA-PC	G^2, 250	Selected repair enforced on 250/250; minimum cost on only 3/111 inconsistent traces (108 higher).	–	Heuristic repair; no exact response threshold.
MPC	G^2, 250	Compatible on 126/139 feasible traces; 13 avoidable gaps; no selected repair.	–	No repair or contestation object.
ASPCR-DAG	native, 250	Full trace on 23/250; graph optimum proved on 250/250.	4330/4330 proved; zero gaps 0/4330; median $\mathrm{gap}/w_f = 2.828$ [0.890,8.647].	Post-hoc graph-objective sensitivity.

Table 6 shifts from availability to the evidence behind a decision. Its G^2 rows use the same 250-trace exact ABA/MPC audit, whereas its ASPCR-DAG row uses a separate 250-run native Bayesian audit. The table therefore compares which objects are certified—minimum repair, enforced selected set, or post-hoc sensitivity—but does not treat the two normalised margin columns as paired effect sizes.

Optimal versus heuristic ABA repair ABA-PC exposes a retained/released fact set and its returned graphs satisfy every retained fact, but its strength-ordered stopping rule computes neither of the category optima $c_{\mathrm{rel}}(f)$ and $c_{\mathrm{keep}}(f)$. Re-running the heuristic with f made mandatory could enforce that fact, but its cost difference would not be the exact margin or inherit Proposition S.1. Table 7 compares the paired repair costs before graph accuracy is considered and, in the final columns, contrasts this repair with MPC's enforcement of the input trace.

On the 312 inconsistent primary traces, the saved OptABA-PC repair costs less on 294 and ties on 18; ABA-PC is never cheaper. The conditional median reduction is 63.4%, with strict improvements on every inconsistent Cancer, Survey, Asia, and SF(8) trace and on 46/47 ER(8) traces. These full-coverage figures include 32 valid timeout incumbents. The exact 250-instance audit in Table 6 proves the corresponding optimum on every pair: among its 111 inconsistent traces, OptABA-PC costs less on 108 and ties on three.

OptABA-PC: fact-level redress and robustness We instantiated the hard-retention contestation and exact margin from Proposition 2, Definition 4, and Proposition S.1 on the frozen OptABA-PC runs. The exact audit uses Cancer, Earthquake, Survey, ER(5), and SF(5), with 50 seeds (2026–2075) per dataset, $N = 5000$, and the matched G^2 test at $\alpha = 0.01$. We do not pool eight-node margins because not every baseline and hard-retention solve reached proved optimality within the computational budget.

For every fact f in the released-fact projection of a baseline optimum, we re-solved the emitted optimisation program after adding the hard constraint that makes f mandatory. Because the baseline optimum releases f, its proved cost gives $c_{\mathrm{rel}}(f)$; the proved hard-retention optimum gives $c_{\mathrm{keep}}(f)$.

Table 7: Repair quality and MPC enforcement on the same 400 primary G^2 traces. Feasible inputs admit a DAG satisfying the entire trace; all other inputs require a repair. *Lower/inconsistent* counts saved OptABA-PC repairs with lower release cost than ABA-PC among the inconsistent inputs, and *same cost* is the complementary count; ABA-PC is never cheaper. Median saving is conditional on a lower saved cost. Thirty-two OptABA-PC rows are valid timeout incumbents; Table 6 separately gives the 250-instance exact audit with proved optima. MPC *full trace* counts returned classes satisfying every input test, and *gap* counts feasible traces that MPC nevertheless fails to satisfy.

	Input status	Saved OptABA-PC repair vs. ABA-PC			MPC output	
Dataset	Feasible / inputs	Lower / inconsistent	Same cost	Median saving	Full trace / inputs	Gap / feasible
Cancer (5)	0/50	50/50	0	40.2%	0/50	–
Earthquake (5)	47/50	1/3	2	61.8%	47/50	0/47
Survey (6)	0/50	50/50	0	57.1%	0/50	–
Asia (8)	0/50	50/50	0	73.0%	0/50	–
ER (5)	16/50	26/34	8	48.4%	11/50	5/16
ER (8)	3/50	46/47	1	66.4%	3/50	0/3
SF (5)	20/50	23/30	7	60.9%	17/50	3/20
SF (8)	2/50	48/48	0	71.6%	2/50	0/2
Total	88/400	**294/312**	18	63.4%	80/400	**8/88**

Table 8: Dataset-level forced-fact evidence over 250 validated baselines per method (50 per dataset). Each target is a fact rejected by the baseline optimum; all re-solves prove optimality and none has zero gap. OptABA-PC's $\mu(f)/w_f$ is an exact user-contestation threshold; c^* denotes the complete baseline optimum repair cost. ASPCR-DAG's $\mu_{\mathrm{ASP}}(f)/w_f$ is instead a post-hoc sensitivity gap for its soft graph-disagreement objective. These are not paired effect sizes: the methods use different traces, weights, and objectives, and their synthetic audits use different graph densities.

	OptABA-PC contestations			ASPCR-DAG sensitivity probes	
Dataset	Proved	Median μ/w_f $[Q_1, Q_3]$	Median μ/c^*	Proved	Median μ_{ASP}/w_f $[Q_1, Q_3]$
Cancer (5)	207/207	0.297 [0.158,0.609]	0.072	959/959	4.041 [1.615,7.946]
Earthquake (5)	4/4	4.170 [1.628,6.332]	3.544	80/80	3.625 [0.990,6.349]
Survey (6)	507/507	0.517 [0.200,1.026]	0.051	2094/2094	2.376 [0.537,14.074]
ER (5)	0/0	–	–	503/503	2.163 [0.773,5.555]
SF (5)	24/24	0.734 [0.308,1.344]	0.182	694/694	2.437 [0.958,6.108]
Total	742/742	0.455 [0.182,0.912]	0.057	4330/4330	2.828 [0.890,8.647]

Their difference therefore directly instantiates the margin in Definition 4. We report $\mu(f)/w_f$ and, to relate this fact-specific threshold to the complete repair, compare $\mu(f)$ with the baseline total release cost $c_{\mathrm{rel}}(f)$.

Table 8 records every dataset-level and pooled threshold. All 742 OptABA-PC contestations prove both category optima, with no timeout, infeasible hard-retention instance, or zero margin; ER(5) has no rejected fact and therefore no imputed value. This instantiates Proposition 2 under its unrestricted-DAG assumptions.

Strict positivity means that no reported release merely selects one member of an already tied retaining/releasing optimum. It also resolves the baseline-witness issue at the fact-selection level: if another baseline optimum retained a contested fact, that fact would have zero margin, while positive weights rule out an optimal strict superset of the reconstructed release. Hence every audited instance has a unique optimal released-fact projection, although it may still have multiple stable-model or graph witnesses.

The margins distinguish releases sensitive to small increases in their cost from those robust to larger increases, holding other costs fixed. For 11.6% of releases, an increase of at most 10% makes retention optimal; 22.5% require more than doubling their release cost (Table 6). These thresholds mark the first tie with a retaining optimum; a strictly larger increase forces every optimum to retain the fact (Proposition S.1). The median margin is 45.5% of the contested fact's release cost; normalising instead by the complete baseline repair cost gives a median of 5.7% (Table 8). Cancer, Survey,

and SF(5) have appreciably different medians, while Earthquake's large estimate rests on only four targets.

Majority-PC: use of tests versus output enforcement A procedure can use a CI test without constraining its final graph to satisfy that test. This distinction is central to interpreting MPC. Within KPC a CI outcome is consumed during skeleton discovery and collider orientation, but is not posted as a global d-separation constraint on the returned CPDAG. Forcing a local test routine would therefore only force that procedural branch; other paths, separating sets, or downstream orientation steps could still yield a Markov class contradicting the same relation. MPC also exposes neither a retained/released fact set nor a global release-cost objective, so $c_{\mathrm{rel}}(f)$ and $c_{\mathrm{keep}}(f)$ are not defined for it.

The final columns of Table 7 report the MPC audit on the same 400 primary traces; the exact 250-trace comparison, including its denser ER(5)/SF(5) setting, appears in Table 6. Since every DAG in one CPDAG has the same conditional-independence model, one consistent extension suffices for the semantic check; endpoint-invalid outputs remain explicit and count as incompatible.

MPC's raw full-compatibility count is dominated by input inconsistency: only 88/400 primary traces are feasible. Conditional on feasibility, it succeeds on 80/88 and leaves eight avoidable gaps, five on ER(5) and three on SF(5). The separate exact audit gives the same conditional pattern: 126/139 compatible feasible traces and 13 gaps, two on ER(5) and 11 on SF(5). The corresponding success rates, 90.9% and 90.6%, show that the change in raw counts reflects trace feasibility and graph density rather than a reversal in method behaviour. MPC can still be a strong graph estimator (Table 11), but it neither identifies a repair on inconsistent inputs nor distinguishes unavoidable contradictions from avoidable output gaps.

ASPCR-DAG: forced-fact sensitivity under a different objective ASPCR-DAG permits a forced-fact sensitivity audit, but not a test of the OptABA-PC contestability guarantee. Let $\mathcal{F}_{\mathrm{ASP}}$ be its exhaustive native Bayesian CI trace and w_i the integer log weight of fact f_i. Its DAG-restricted encoding derives $\mathsf{fail}(f_i)$ exactly when a candidate graph contradicts f_i and minimises

$$C_{\mathrm{ASP}}(G) = \sum_i w_i \, \mathbf{1}[G \not\models f_i].$$

The term $\mathbf{1}[\cdot]$ denotes the indicator function, equal to one when its argument is true and zero otherwise. For every fact f failed by the saved optimum, we perform what we call a *hardening*: as a sensitivity probe, we add the integrity constraint $\leftarrow \mathsf{fail}(f)$ and re-solve. This turns that one test from a soft, penalised correspondence into a mandatory requirement that the new DAG satisfy it, while every other test remains soft. Hence

$$\mu_{\mathrm{ASP}}(f) = \min_{G \models f} C_{\mathrm{ASP}}(G) - \min_G C_{\mathrm{ASP}}(G)$$

is an exact sensitivity margin for the modified ASPCR-DAG objective: the new DAG is forced to correspond to the targeted test, while its correspondence to every other test remains soft. The hard constraint is imposed by our audit rather than supplied as a method-level redress operation, and ASPCR-DAG has no counterpart of Proposition S.1. It returns one optimal DAG rather than the complete structural set supported by an explicit correction set.

Tables 6 and 8 report this diagnostic alongside, but separately from, the OptABA-PC audit. The regenerated ASP programs reproduce the saved proved objective on all 250 ASPCR-DAG baselines. Only 23 native traces have zero graph–test disagreement; the other baselines supply 4,330 failed facts for hardening. Every hardening reaches a proved optimum, with no timeout, infeasible target, enforcement failure, or zero gap. Thus every graph at the original optimum cost contradicts the targeted test; enforcing correspondence to it changes the optimum but leaves all remaining test–graph correspondences soft. This is sensitivity evidence about ASPCR-DAG's objective, not actual redress since no consistent repair is offered.

Within that objective, the pooled median normalised gap is 2.828, with the dataset variation shown in Table 8. If $g = \mu_{\mathrm{ASP}}(f)/w_f$, hardening removes the target's penalty w_f yet raises the optimum by gw_f; the net penalty transferred to other disagreements is therefore $(1+g)w_f$. At the pooled median this is 3.828 times the target's weight. This is useful sensitivity information, but not evidence

Table 9: Absolute retained-fact quality and runtime for the primary ABA-PC–OptABA-PC comparison (mean ± standard deviation over 50 matched seeds, $N = 5000$, $\alpha = 0.01$). Both methods repair the same recorded CI tests. All eight datasets are included, with node counts in brackets. Correct retained counts facts agreeing with d-separation in the generating DAG; Fact F1 scores retained-fact classification. Time includes program construction, solving, and compatible-set evaluation. Arrows indicate preferred directions. Only a significant paired ABA-PC–OptABA-PC winner is bold and starred: one, two, and three stars denote BH-adjusted $p < 0.05$, 0.01, and 0.001 from two-sided Wilcoxon tests. CPDAG evaluation is in Table 10.

Dataset	Method	Correct retained ↑	Fact F1 ↑	Time (s) ↓
Cancer (5)	ABA-PC	24.0 ± 8.7	0.904 ± 0.046	1.51 ± 0.48
	OptABA-PC	$\mathbf{25.6 \pm 9.8}^{***}$	0.900 ± 0.064	$\mathbf{1.28 \pm 0.38}^{***}$
Earthquake (5)	ABA-PC	54.1 ± 0.8	0.999 ± 0.007	1.83 ± 0.83
	OptABA-PC	54.2 ± 0.8	0.999 ± 0.003	$\mathbf{1.68 \pm 0.81}^{***}$
Survey (6)	ABA-PC	24.8 ± 9.6	$\mathbf{0.789 \pm 0.075}^{**}$	15.7 ± 30.2
	OptABA-PC	$\mathbf{29.7 \pm 12.3}^{***}$	0.742 ± 0.107	2.94 ± 1.75
Asia (8)	ABA-PC	105.8 ± 8.8	0.891 ± 0.018	95.4 ± 48.8
	OptABA-PC	$\mathbf{127.6 \pm 9.7}^{***}$	$\mathbf{0.973 \pm 0.015}^{***}$	$\mathbf{66.5 \pm 84.9}^{*}$
ER (5)	ABA-PC	30.9 ± 7.3	0.953 ± 0.049	1.01 ± 0.15
	OptABA-PC	$\mathbf{32.2 \pm 7.3}^{***}$	$\mathbf{0.969 \pm 0.039}^{**}$	$\mathbf{0.90 \pm 0.08}^{***}$
ER (8)	ABA-PC	71.4 ± 23.5	0.871 ± 0.069	$\mathbf{48.0 \pm 54.3}^{***}$
	OptABA-PC	$\mathbf{86.8 \pm 25.7}^{***}$	$\mathbf{0.953 \pm 0.043}^{***}$	139.1 ± 139.4
SF (5)	ABA-PC	29.0 ± 7.2	0.953 ± 0.056	1.85 ± 2.03
	OptABA-PC	$\mathbf{30.7 \pm 6.0}^{***}$	$\mathbf{0.987 \pm 0.026}^{***}$	$\mathbf{1.76 \pm 2.05}^{***}$
SF (8)	ABA-PC	72.1 ± 37.1	0.838 ± 0.065	$\mathbf{42.3 \pm 53.4}^{**}$
	OptABA-PC	$\mathbf{93.2 \pm 45.4}^{***}$	$\mathbf{0.931 \pm 0.051}^{***}$	99.1 ± 122.0

that ASPCR-DAG is "more contestable": its weights, trace, objective, and operation differ from OptABA-PC's.

The absence of zero gaps agrees only at the level of solver sensitivity: neither saved optimum fails a targeted fact merely because it selected one member of a retaining/failing tie. It does not transfer OptABA-PC's correction-set semantics, total fact-enforcement result, or exact response guarantee to ASPCR-DAG. Nor should the normalised values be ranked: ASPCR-DAG uses all $\binom{n}{2}2^{n-2}$ native Bayesian tests and log weights, whereas OptABA-PC uses the matched adaptive G^2 trace and corresponding release weights.

G.4 Additional Empirical Results and Discussion

Tables 9–18 give the complete evidence behind the compact primary comparison for the fixed bnlearn networks and the ER/SF graphs under the setup in the main paper. Tables 9 and 10 report absolute ABA-PC–OptABA-PC fact/time and CPDAG results for all eight datasets, including Earthquake; Table 12 gives the fact counts and Table 13 their precision, recall, and F1; Table 11 compares graph-level results; Tables 14–17 give SHD and F1 reconstruction ranges; and Table 18 reports timing, output-set sizes, and CI compatibility. Values are absolute mean ± standard deviation. Stars compare OptABA-PC with ABA-PC; daggers compare an all other baselines with OptABA-PC. No missing value is imputed. ASPCR-DAG is reported on Cancer, Earthquake, Survey, ER(5), and SF(5), with 50 matched runs on each.

Graph-level findings. Table 11 confirms that optimal repair improves mean CPDAG reconstruction over heuristic repair on seven datasets, while Earthquake is effectively tied. Across the five matched ASPCR-DAG datasets, OptABA-PC has significantly higher mean CPDAG F1 throughout and lower SHD on four; Survey SHD is indistinguishable. Its returned-DAG F1 is likewise higher on all five and SHD on four; SF(5) DAG SHD is indistinguishable. Native discrete FGS is a strong graph estimator: it significantly outperforms OptABA-PC on both mean CPDAG metrics for Cancer, ER(5), ER(8), and SF(5), on F1 for Survey, and on SHD for SF(8). OptABA-PC significantly outperforms FGS on both metrics for Earthquake and on SHD for Asia; the remaining mean CPDAG differences are not significant. This mixed comparison reinforces the distinction between

Table 10: Absolute CPDAG evaluation for the primary ABA-PC–OptABA-PC comparison on the same 50 matched seeds per dataset as Table 9. Counts, F1, and SHD are mean ± standard deviation. Node counts are in brackets. #CPDAGs counts saved equivalence classes satisfying every retained post-repair fact; F1 and SHD average reconstruction quality over these budget-limited sets. Arrows indicate preferred directions. Only a significant paired ABA-PC–OptABA-PC winner is bold and starred: one, two, and three stars denote BH-adjusted $p < 0.05$, 0.01, and 0.001 from two-sided Wilcoxon tests.

Dataset	Method	#CPDAGs ↓	CPDAG F1 ↑	CPDAG SHD ↓
Cancer (5)	ABA-PC	15.1 ± 28.8	0.522 ± 0.214	2.81 ± 1.18
	OptABA-PC	$\mathbf{1.0 \pm 0.0}^{***}$	$\mathbf{0.576 \pm 0.259}^{*}$	$\mathbf{1.84 \pm 1.11}^{***}$
Earthquake (5)	ABA-PC	1.0 ± 0.3	0.991 ± 0.033	0.09 ± 0.30
	OptABA-PC	1.0 ± 0.0	0.991 ± 0.030	0.08 ± 0.27
Survey (6)	ABA-PC	1278.5 ± 2807.2	0.438 ± 0.088	7.14 ± 1.39
	OptABA-PC	$\mathbf{1.2 \pm 0.7}^{***}$	$\mathbf{0.597 \pm 0.127}^{***}$	$\mathbf{4.35 \pm 1.09}^{***}$
Asia (8)	ABA-PC	10822.7 ± 7112.3	0.224 ± 0.073	14.4 ± 3.26
	OptABA-PC	$\mathbf{1.2 \pm 0.4}^{***}$	$\mathbf{0.536 \pm 0.061}^{***}$	$\mathbf{3.86 \pm 0.60}^{***}$
ER (5)	ABA-PC	8.2 ± 13.0	0.562 ± 0.224	2.88 ± 2.01
	OptABA-PC	$\mathbf{3.0 \pm 3.4}^{***}$	$\mathbf{0.589 \pm 0.213}^{*}$	$\mathbf{2.09 \pm 1.71}^{**}$
ER (8)	ABA-PC	3840.0 ± 5377.9	0.478 ± 0.179	8.19 ± 4.21
	OptABA-PC	$\mathbf{4.7 \pm 9.5}^{***}$	$\mathbf{0.627 \pm 0.167}^{***}$	$\mathbf{3.53 \pm 1.89}^{***}$
SF (5)	ABA-PC	88.4 ± 202.7	0.670 ± 0.176	2.77 ± 2.10
	OptABA-PC	$\mathbf{85.1 \pm 203.9}^{**}$	$\mathbf{0.737 \pm 0.148}^{**}$	$\mathbf{2.03 \pm 1.97}^{**}$
SF (8)	ABA-PC	3108.1 ± 4287.4	0.498 ± 0.154	8.15 ± 3.77
	OptABA-PC	$\mathbf{170.3 \pm 1023.2}^{***}$	$\mathbf{0.629 \pm 0.129}^{***}$	$\mathbf{4.42 \pm 2.05}^{***}$

reconstruction and repair: FGS selects a graph by score but provides no retained CI set, compatibility audit, or contestation object. The ASPCR-DAG comparisons use the same saved samples and generating DAG, but their different fact universes, weights, and optimisation costs are not numerically comparable. MPC is stronger on both mean CPDAG metrics for Survey and the four synthetic datasets; Cancer favours MPC only on CPDAG F1, Asia favours OptABA-PC on CPDAG F1, and Earthquake is tied. These are graph-estimation comparisons, not fact-repair comparisons.

The synthetic MPC result is consistent with Majority-PC's specialisation. It returns one targeted CPDAG when structurally valid, whose reversible orientations remain uncommitted, whereas OptABA-PC is evaluated over the saved, budget-limited structural set admitted by its selected coherent fact repair. OptABA-PC admits on average 3.0, 4.7, 85.1, and 170.3 compatible CPDAGs on ER(5), ER(8), SF(5), and SF(8), respectively (Table 11). MPC beats the OptABA-PC average on all four, but does not significantly beat its best compatible CPDAG on ER(5), ER(8), or SF(5); SF(8) remains a significant MPC advantage (Tables 14 and 15). This narrowing should not be mistaken for a best-DAG win by OptABA-PC. After expanding each valid class, MPC has significantly better best-DAG SHD and F1 on ER(5), SF(5), and SF(8), and better best-DAG SHD on ER(8), where best-DAG F1 is indistinguishable (Tables 16 and 17). Thus comparing MPC's single partially oriented class with OptABA-PC's saved output set explains part, but not all, of MPC's average margin. In this broader sparse primary setting, full-trace compatibility holds for only 22%, 6%, 34%, and 4% of the corresponding MPC outputs (Table 18); Table 6 gives the separate exact 250-trace comparison. Its structural advantage therefore does not establish a fact-repair advantage.

The range tables further show that OptABA-PC significantly improves worst CPDAG and worst DAG SHD/F1 over ABA-PC on seven of eight datasets. Best-of-set values can favour ABA-PC because its much larger output set has more opportunities to contain one unusually accurate member; average and worst values expose the cost of that ambiguity. Compatibility is stricter than structural validity: across the broader eight-dataset primary setting, only 80 of 400 MPC outputs have a CPDAG satisfying every recorded G^2 outcome, while Cancer, Survey, and Asia have none (Table 18). For saved ASPCR-DAG outputs the corresponding native-trace count is 32 of 250: 22 on Earthquake, six on ER(5), and four on SF(5). FGS has no CI trace and is therefore marked – rather than zero.

Table 11: Graph reconstruction and CI-compatible equivalence classes on the matched primary experiments (mean ± standard deviation over seeds). Numbers in brackets after dataset names give the number of nodes. ABA-PC and OptABA-PC CPDAG values average over every equivalence class admitted by the selected repair; MPC and FGS return one CPDAG when their output is structurally valid, and ASPCR-DAG returns one DAG whose equivalence class supplies its CPDAG score. DAG columns average every compatible DAG for ABA-PC/OptABA-PC and every consistent extension of a valid MPC/FGS CPDAG; ASPCR-DAG is scored on its returned DAG. The final column counts CPDAGs satisfying the enforced evidence: retained post-repair facts for ABA-PC/OptABA-PC and every recorded test for MPC/ASPCR-DAG; it is undefined for score-based FGS ('–'). Stars identify the significant ABA-PC–OptABA-PC winner. Daggers on an external method indicate a paired difference from OptABA-PC; bold daggers favour that external method, whereas unbold daggers favour OptABA-PC. One, two, and three symbols denote BH-adjusted $p < 0.05$, 0.01, and 0.001. Subscripts occur only for MPC DAG-extension metrics when its endpoint output is invalid: $n = 38$ on Survey and $n = 49$ on ER(8).

Dataset	Method	CPDAG SHD ↓	CPDAG F1 ↑	DAG SHD ↓	DAG F1 ↑	Compat. CPDAGs ↓
Cancer (5)	ABA-PC	2.81 ± 1.18	0.522 ± 0.214	2.98 ± 1.20	0.483 ± 0.223	15.1 ± 28.8
	OptABA-PC	$\mathbf{1.84 \pm 1.11}^{***}$	$\mathbf{0.576 \pm 0.259}^{*}$	$\mathbf{1.89 \pm 1.14}^{***}$	$\mathbf{0.563 \pm 0.262}^{*}$	$\mathbf{1.0 \pm 0.0}^{***}$
	MPC	1.58 ± 0.97	$\mathbf{0.661 \pm 0.202}^{\dagger}$	1.70 ± 1.01	0.623 ± 0.222	0.0 ± 0.0
	FGS	$\mathbf{1.12 \pm 0.96}^{\dagger\dagger}$	$\mathbf{0.742 \pm 0.212}^{\dagger\dagger}$	$\mathbf{1.11 \pm 0.94}^{\dagger\dagger\dagger}$	$\mathbf{0.739 \pm 0.215}^{\dagger\dagger}$	–
	ASPCR-DAG	$2.24 \pm 0.56^{\dagger}$	$0.490 \pm 0.123^{\dagger}$	$2.34 \pm 0.89^{\dagger}$	$0.462 \pm 0.225^{\dagger}$	0.0 ± 0.0
Earthquake (5)	ABA-PC	0.09 ± 0.30	0.991 ± 0.033	0.10 ± 0.34	0.989 ± 0.039	1.0 ± 0.3
	OptABA-PC	0.08 ± 0.27	0.991 ± 0.030	0.08 ± 0.27	0.991 ± 0.030	1.0 ± 0.0
	MPC	0.20 ± 0.73	0.991 ± 0.030	0.08 ± 0.27	0.991 ± 0.030	0.9 ± 0.2
	FGS	$4.64 \pm 1.34^{\dagger\dagger\dagger}$	$0.422 \pm 0.143^{\dagger\dagger\dagger}$	$4.41 \pm 1.15^{\dagger\dagger\dagger}$	$0.417 \pm 0.149^{\dagger\dagger\dagger}$	–
	ASPCR-DAG	$1.08 \pm 1.16^{\dagger\dagger\dagger}$	$0.866 \pm 0.168^{\dagger\dagger\dagger}$	$1.06 \pm 1.20^{\dagger\dagger\dagger}$	$0.871 \pm 0.176^{\dagger\dagger\dagger}$	0.4 ± 0.5
Survey (6)	ABA-PC	7.14 ± 1.39	0.438 ± 0.088	6.73 ± 1.39	0.340 ± 0.103	1278.5 ± 2807.2
	OptABA-PC	$\mathbf{4.35 \pm 1.09}^{***}$	$\mathbf{0.597 \pm 0.127}^{***}$	$\mathbf{3.77 \pm 0.91}^{***}$	$\mathbf{0.482 \pm 0.140}^{***}$	$\mathbf{1.2 \pm 0.7}^{***}$
	MPC	$\mathbf{3.68 \pm 1.49}^{\dagger\dagger}$	$\mathbf{0.815 \pm 0.088}^{\dagger\dagger\dagger}$	$\mathbf{3.17 \pm 1.49}^{\dagger}_{38}$	$0.528 \pm 0.242_{38}$	0.0 ± 0.0
	FGS	3.88 ± 1.55	$\mathbf{0.708 \pm 0.187}^{\dagger\dagger\dagger}$	3.47 ± 1.28	0.465 ± 0.218	–
	ASPCR-DAG	4.52 ± 1.23	$0.496 \pm 0.154^{\dagger\dagger}$	$4.16 \pm 1.23^{\dagger}$	$0.385 \pm 0.220^{\dagger\dagger}$	0.0 ± 0.0
Asia (8)	ABA-PC	14.4 ± 3.26	0.224 ± 0.073	14.0 ± 2.81	0.310 ± 0.095	10822.7 ± 7112.3
	OptABA-PC	$\mathbf{3.86 \pm 0.60}^{***}$	$\mathbf{0.536 \pm 0.061}^{***}$	$\mathbf{4.51 \pm 0.58}^{***}$	$\mathbf{0.665 \pm 0.009}^{***}$	$\mathbf{1.2 \pm 0.4}^{***}$
	MPC	3.82 ± 0.44	$0.483 \pm 0.045^{\dagger\dagger\dagger}$	$\mathbf{3.92 \pm 0.25}^{\dagger\dagger\dagger}$	$0.620 \pm 0.013^{\dagger\dagger\dagger}$	0.0 ± 0.0
	FGS	$6.66 \pm 2.22^{\dagger\dagger\dagger}$	0.561 ± 0.102	$6.01 \pm 1.70^{\dagger\dagger\dagger}$	$0.610 \pm 0.105^{\dagger\dagger}$	–
ER (5)	ABA-PC	2.88 ± 2.01	0.562 ± 0.224	3.17 ± 1.73	0.567 ± 0.215	8.2 ± 13.0
	OptABA-PC	$\mathbf{2.09 \pm 1.71}^{**}$	$\mathbf{0.589 \pm 0.213}^{*}$	$\mathbf{2.49 \pm 1.35}^{***}$	$\mathbf{0.620 \pm 0.188}^{**}$	$\mathbf{3.0 \pm 3.4}^{***}$
	MPC	$\mathbf{0.96 \pm 1.32}^{\dagger\dagger}$	$\mathbf{0.712 \pm 0.251}^{\dagger\dagger\dagger}$	$\mathbf{1.23 \pm 0.94}^{\dagger\dagger\dagger}$	$\mathbf{0.768 \pm 0.177}^{\dagger\dagger\dagger}$	0.2 ± 0.4
	FGS	$\mathbf{1.14 \pm 1.44}^{\dagger\dagger}$	$\mathbf{0.677 \pm 0.239}^{\dagger\dagger}$	$\mathbf{1.58 \pm 1.11}^{\dagger\dagger\dagger}$	$\mathbf{0.710 \pm 0.196}^{\dagger\dagger}$	–
	ASPCR-DAG	$2.90 \pm 1.25^{\dagger\dagger}$	$0.377 \pm 0.292^{\dagger\dagger\dagger}$	$3.26 \pm 1.19^{\dagger\dagger}$	$0.426 \pm 0.251^{\dagger\dagger\dagger}$	0.1 ± 0.3
ER (8)	ABA-PC	8.19 ± 4.21	0.478 ± 0.179	7.99 ± 3.77	0.496 ± 0.181	3840.0 ± 5377.9
	OptABA-PC	$\mathbf{3.53 \pm 1.89}^{***}$	$\mathbf{0.627 \pm 0.167}^{***}$	$\mathbf{3.85 \pm 1.58}^{***}$	$\mathbf{0.650 \pm 0.137}^{***}$	$\mathbf{4.7 \pm 9.5}^{***}$
	MPC	$\mathbf{2.12 \pm 1.79}^{\dagger\dagger\dagger}$	$\mathbf{0.704 \pm 0.137}^{\dagger\dagger}$	$\mathbf{2.38 \pm 1.21}^{\dagger\dagger\dagger}_{49}$	$\mathbf{0.727 \pm 0.140}^{\dagger\dagger}_{49}$	0.1 ± 0.2
	FGS	$\mathbf{1.56 \pm 1.99}^{\dagger\dagger\dagger}$	$\mathbf{0.744 \pm 0.157}^{\dagger\dagger\dagger}$	$\mathbf{2.07 \pm 1.41}^{\dagger\dagger\dagger}$	$\mathbf{0.774 \pm 0.139}^{\dagger\dagger\dagger}$	–
SF (5)	ABA-PC	2.77 ± 2.10	0.670 ± 0.176	3.07 ± 1.94	0.594 ± 0.213	88.4 ± 202.7
	OptABA-PC	$\mathbf{2.03 \pm 1.97}^{**}$	$\mathbf{0.737 \pm 0.148}^{**}$	$\mathbf{2.52 \pm 1.89}^{**}$	$\mathbf{0.650 \pm 0.197}^{**}$	$\mathbf{85.1 \pm 203.9}^{**}$
	MPC	$\mathbf{0.34 \pm 0.80}^{\dagger\dagger\dagger}$	$\mathbf{0.874 \pm 0.173}^{\dagger\dagger\dagger}$	$\mathbf{0.48 \pm 0.58}^{\dagger\dagger\dagger}$	$\mathbf{0.893 \pm 0.120}^{\dagger\dagger\dagger}$	0.3 ± 0.5
	FGS	$\mathbf{0.56 \pm 0.93}^{\dagger\dagger\dagger}$	$\mathbf{0.854 \pm 0.173}^{\dagger\dagger\dagger}$	$\mathbf{0.69 \pm 0.85}^{\dagger\dagger\dagger}$	$\mathbf{0.861 \pm 0.143}^{\dagger\dagger\dagger}$	–
	ASPCR-DAG	$2.76 \pm 1.51^{\dagger}$	$0.448 \pm 0.303^{\dagger\dagger\dagger}$	2.60 ± 1.26	$0.448 \pm 0.283^{\dagger\dagger\dagger}$	0.1 ± 0.3
SF (8)	ABA-PC	8.15 ± 3.77	0.498 ± 0.154	8.14 ± 3.46	0.481 ± 0.166	3108.1 ± 4287.4
	OptABA-PC	$\mathbf{4.42 \pm 2.05}^{***}$	$\mathbf{0.629 \pm 0.129}^{***}$	$\mathbf{4.76 \pm 1.97}^{***}$	$\mathbf{0.601 \pm 0.146}^{***}$	$\mathbf{170.3 \pm 1023.2}^{***}$
	MPC	$\mathbf{1.58 \pm 1.49}^{\dagger\dagger\dagger}$	$\mathbf{0.774 \pm 0.163}^{\dagger\dagger\dagger}$	$\mathbf{1.88 \pm 1.23}^{\dagger\dagger\dagger}$	$\mathbf{0.773 \pm 0.143}^{\dagger\dagger\dagger}$	0.0 ± 0.2
	FGS	$\mathbf{2.50 \pm 1.96}^{\dagger\dagger\dagger}$	0.683 ± 0.246	$\mathbf{2.82 \pm 1.88}^{\dagger\dagger\dagger}$	0.664 ± 0.241	–

Fact quality and computation. Table 12 reports retained correct/false facts; Table 13 reports their precision, recall, and FactF1. Table 18 reports timing and structural and CI-compatible output counts. Paired ABA-PC–OptABA-PC timings include program construction, solving, and compatible-set evaluation on a like-for-like basis. Cross-method scopes differ: MPC/FGS time the learner call, while ASPCR-DAG reports its native end-to-end R run. They describe implementation cost, not a controlled speed ranking.

Table 12: CI-fact counts on the matched primary experiments (mean ± standard deviation over 50 seeds per reported row). Node counts are in brackets. Correct available counts recorded tests agreeing with d-separation in the generating DAG; correct/false kept split retained facts. ABA-PC and OptABA-PC repair the identical test set. Only a significant paired ABA-PC–OptABA-PC winner is bold and starred: one, two, and three stars denote BH-adjusted $p < 0.05$, 0.01, and 0.001 from two-sided Wilcoxon tests. For ASPCR-DAG, *kept* means its returned DAG satisfies a native soft test relation: descriptive graph–test agreement, not an explicit retained CI set. MPC and FGS expose no comparable fact accounting. Precision, recall, and F1 are in Table 13.

Dataset	Method	Correct available	Correct kept	False kept
Cancer (5)	ABA-PC	27.2 ± 10.1	24.0 ± 8.7	$\mathbf{1.6 \pm 1.5}^{***}$
	OptABA-PC	27.2 ± 10.1	$\mathbf{25.6 \pm 9.8}^{***}$	3.3 ± 2.2
	ASPCR-DAG	53.7 ± 3.4	48.0 ± 7.2	12.9 ± 8.5
Earthquake (5)	ABA-PC	54.2 ± 0.9	54.1 ± 0.8	0.1 ± 0.3
	OptABA-PC	54.2 ± 0.9	54.2 ± 0.8	0.1 ± 0.3
	ASPCR-DAG	75.6 ± 2.9	74.7 ± 3.8	3.7 ± 3.0
Survey (6)	ABA-PC	34.9 ± 13.2	24.8 ± 9.6	$\mathbf{2.4 \pm 1.5}^{***}$
	OptABA-PC	34.9 ± 13.2	$\mathbf{29.7 \pm 12.3}^{***}$	13.4 ± 4.7
	ASPCR-DAG	135.0 ± 11.2	116.1 ± 12.8	82.0 ± 18.5
Asia (8)	ABA-PC	129.2 ± 9.6	105.8 ± 8.8	$\mathbf{2.4 \pm 0.9}^{***}$
	OptABA-PC	129.2 ± 9.6	$\mathbf{127.6 \pm 9.7}^{***}$	5.4 ± 3.1
ER (5)	ABA-PC	32.7 ± 7.3	30.9 ± 7.3	$\mathbf{1.2 \pm 1.6}^{**}$
	OptABA-PC	32.7 ± 7.3	$\mathbf{32.2 \pm 7.3}^{***}$	1.5 ± 1.9
	ASPCR-DAG	52.1 ± 9.9	47.3 ± 10.1	22.3 ± 11.8
ER (8)	ABA-PC	89.2 ± 26.2	71.4 ± 23.5	$\mathbf{2.7 \pm 2.6}^{***}$
	OptABA-PC	89.2 ± 26.2	$\mathbf{86.8 \pm 25.7}^{***}$	6.0 ± 5.8
SF (5)	ABA-PC	30.9 ± 5.8	29.0 ± 7.2	0.5 ± 0.6
	OptABA-PC	30.9 ± 5.8	$\mathbf{30.7 \pm 6.0}^{***}$	0.5 ± 0.9
	ASPCR-DAG	49.0 ± 9.7	42.9 ± 9.4	27.0 ± 13.2
SF (8)	ABA-PC	96.6 ± 45.5	72.1 ± 37.1	$\mathbf{2.6 \pm 2.8}^{***}$
	OptABA-PC	96.6 ± 45.5	$\mathbf{93.2 \pm 45.4}^{***}$	9.0 ± 7.3

Table 13: CI-fact precision, recall, and F1 on the matched primary experiments (mean ± standard deviation over 50 seeds per reported row; higher is better). Node counts are in brackets. Retained correct tests are true positives, retained false tests are false positives, and released correct tests are false negatives; counts are in Table 12. For ASPCR-DAG, retention denotes satisfaction of native soft tests by its returned DAG, not an explicit retained CI set. MPC and FGS expose no comparable fact accounting. Only a significant paired ABA-PC–OptABA-PC winner is bold and starred: one, two, and three stars denote BH-adjusted $p < 0.05$, 0.01, and 0.001 from two-sided Wilcoxon tests.

Dataset	Method	Precision	Recall	Fact F1
Cancer (5)	ABA-PC	$\mathbf{0.927 \pm 0.069}^{***}$	0.887 ± 0.058	0.904 ± 0.046
	OptABA-PC	0.866 ± 0.091	$\mathbf{0.941 \pm 0.045}^{***}$	0.900 ± 0.064
	ASPCR-DAG	0.792 ± 0.132	0.892 ± 0.114	0.837 ± 0.120
Earthquake (5)	ABA-PC	0.999 ± 0.005	0.999 ± 0.010	0.999 ± 0.007
	OptABA-PC	0.999 ± 0.005	1.000 ± 0.002	0.999 ± 0.003
	ASPCR-DAG	0.952 ± 0.039	0.987 ± 0.018	0.969 ± 0.026
Survey (6)	ABA-PC	$\mathbf{0.894 \pm 0.088}^{***}$	0.716 ± 0.105	$\mathbf{0.789 \pm 0.075}^{**}$
	OptABA-PC	0.669 ± 0.126	$\mathbf{0.842 \pm 0.082}^{***}$	0.742 ± 0.107
	ASPCR-DAG	0.589 ± 0.077	0.862 ± 0.082	0.697 ± 0.070
Asia (8)	ABA-PC	$\mathbf{0.978 \pm 0.008}^{***}$	0.819 ± 0.032	0.891 ± 0.018
	OptABA-PC	0.959 ± 0.023	$\mathbf{0.988 \pm 0.009}^{***}$	$\mathbf{0.973 \pm 0.015}^{***}$
ER (5)	ABA-PC	$\mathbf{0.963 \pm 0.051}^{*}$	0.946 ± 0.064	0.953 ± 0.049
	OptABA-PC	0.955 ± 0.056	$\mathbf{0.984 \pm 0.027}^{***}$	$\mathbf{0.969 \pm 0.039}^{**}$
	ASPCR-DAG	0.684 ± 0.155	0.907 ± 0.085	0.770 ± 0.116
ER (8)	ABA-PC	$\mathbf{0.962 \pm 0.031}^{***}$	0.800 ± 0.104	0.871 ± 0.069
	OptABA-PC	0.934 ± 0.055	$\mathbf{0.973 \pm 0.033}^{***}$	$\mathbf{0.953 \pm 0.043}^{***}$
SF (5)	ABA-PC	0.982 ± 0.025	0.929 ± 0.090	0.953 ± 0.056
	OptABA-PC	0.982 ± 0.032	$\mathbf{0.992 \pm 0.022}^{***}$	$\mathbf{0.987 \pm 0.026}^{***}$
	ASPCR-DAG	0.623 ± 0.169	0.880 ± 0.108	0.717 ± 0.128
SF (8)	ABA-PC	$\mathbf{0.957 \pm 0.045}^{***}$	0.752 ± 0.106	0.838 ± 0.065
	OptABA-PC	0.903 ± 0.066	$\mathbf{0.962 \pm 0.043}^{***}$	$\mathbf{0.931 \pm 0.051}^{***}$

Table 14: Average, best, and worst CPDAG SHD within each run of the matched primary experiments (mean ± standard deviation over seeds; lower is better). ABA-PC and OptABA-PC are scored over the saved classes satisfying their selected facts. MPC and FGS return one class when structurally valid; ASPCR-DAG's DAG supplies one class, so their average, best, and worst values coincide within a run. Stars mark the significant ABA-PC–OptABA-PC winner. Daggers compare an external method with OptABA-PC; bold daggers favour that method, unbold daggers favour OptABA-PC. One, two, and three symbols denote BH-adjusted $p < 0.05$, 0.01, and 0.001. Compatibility counts are in Table 18.

Dataset	Method	SHD avg	SHD best	SHD worst
Cancer (5)	ABA-PC	2.81 ± 1.18	1.58 ± 1.18	3.84 ± 1.80
	OptABA-PC	$\mathbf{1.84 \pm 1.11}^{***}$	1.84 ± 1.11	$\mathbf{1.84 \pm 1.11}^{***}$
	MPC	1.58 ± 0.97	1.58 ± 0.97	1.58 ± 0.97
	FGS	$\mathbf{1.12 \pm 0.96}^{\dagger\dagger}$	$\mathbf{1.12 \pm 0.96}^{\dagger\dagger}$	$\mathbf{1.12 \pm 0.96}^{\dagger\dagger}$
	ASPCR-DAG	$2.24 \pm 0.56^{\dagger}$	$2.24 \pm 0.56^{\dagger}$	$2.24 \pm 0.56^{\dagger}$
Earthquake (5)	ABA-PC	0.09 ± 0.30	0.08 ± 0.27	0.10 ± 0.36
	OptABA-PC	0.08 ± 0.27	0.08 ± 0.27	0.08 ± 0.27
	MPC	0.20 ± 0.73	0.20 ± 0.73	0.20 ± 0.73
	FGS	$4.64 \pm 1.34^{\dagger\dagger\dagger}$	$4.64 \pm 1.34^{\dagger\dagger\dagger}$	$4.64 \pm 1.34^{\dagger\dagger\dagger}$
	ASPCR-DAG	$1.08 \pm 1.16^{\dagger\dagger\dagger}$	$1.08 \pm 1.16^{\dagger\dagger\dagger}$	$1.08 \pm 1.16^{\dagger\dagger\dagger}$
Survey (6)	ABA-PC	7.14 ± 1.39	$\mathbf{3.18 \pm 1.66}^{**}$	9.96 ± 2.34
	OptABA-PC	$\mathbf{4.35 \pm 1.09}^{***}$	4.26 ± 1.08	$\mathbf{4.44 \pm 1.16}^{***}$
	MPC	$\mathbf{3.68 \pm 1.49}^{\dagger\dagger}$	$\mathbf{3.68 \pm 1.49}^{\dagger}$	$\mathbf{3.68 \pm 1.49}^{\dagger\dagger}$
	FGS	3.88 ± 1.55	3.88 ± 1.55	3.88 ± 1.55
	ASPCR-DAG	4.52 ± 1.23	4.52 ± 1.23	4.52 ± 1.23
Asia (8)	ABA-PC	14.4 ± 3.26	8.26 ± 2.59	20.0 ± 4.20
	OptABA-PC	$\mathbf{3.86 \pm 0.60}^{***}$	$\mathbf{3.76 \pm 0.69}^{***}$	$\mathbf{3.96 \pm 0.57}^{***}$
	MPC	3.82 ± 0.44	3.82 ± 0.44	3.82 ± 0.44
	FGS	$6.66 \pm 2.22^{\dagger\dagger\dagger}$	$6.66 \pm 2.22^{\dagger\dagger\dagger}$	$6.66 \pm 2.22^{\dagger\dagger\dagger}$
ER (5)	ABA-PC	2.88 ± 2.01	1.68 ± 1.96	3.88 ± 2.48
	OptABA-PC	$\mathbf{2.09 \pm 1.71}^{**}$	1.34 ± 1.60	$\mathbf{2.72 \pm 2.14}^{**}$
	MPC	$\mathbf{0.96 \pm 1.32}^{\dagger\dagger}$	0.96 ± 1.32	$\mathbf{0.96 \pm 1.32}^{\dagger\dagger\dagger}$
	FGS	$\mathbf{1.14 \pm 1.44}^{\dagger\dagger}$	1.14 ± 1.44	$\mathbf{1.14 \pm 1.44}^{\dagger\dagger\dagger}$
	ASPCR-DAG	$2.90 \pm 1.25^{\dagger\dagger}$	$2.90 \pm 1.25^{\dagger\dagger\dagger}$	2.90 ± 1.25
ER (8)	ABA-PC	8.19 ± 4.21	3.72 ± 2.88	12.2 ± 6.21
	OptABA-PC	$\mathbf{3.53 \pm 1.89}^{***}$	$\mathbf{2.74 \pm 1.98}^{*}$	$\mathbf{4.26 \pm 2.16}^{***}$
	MPC	$\mathbf{2.12 \pm 1.79}^{\dagger\dagger\dagger}$	2.12 ± 1.79	$\mathbf{2.12 \pm 1.79}^{\dagger\dagger\dagger}$
	FGS	$\mathbf{1.56 \pm 1.99}^{\dagger\dagger\dagger}$	$\mathbf{1.56 \pm 1.99}^{\dagger\dagger}$	$\mathbf{1.56 \pm 1.99}^{\dagger\dagger\dagger}$
SF (5)	ABA-PC	2.77 ± 2.10	1.04 ± 1.80	4.56 ± 3.36
	OptABA-PC	$\mathbf{2.03 \pm 1.97}^{**}$	$\mathbf{0.60 \pm 1.46}^{*}$	$\mathbf{3.54 \pm 3.39}^{**}$
	MPC	$\mathbf{0.34 \pm 0.80}^{\dagger\dagger\dagger}$	0.34 ± 0.80	$\mathbf{0.34 \pm 0.80}^{\dagger\dagger\dagger}$
	FGS	$\mathbf{0.56 \pm 0.93}^{\dagger\dagger\dagger}$	0.56 ± 0.93	$\mathbf{0.56 \pm 0.93}^{\dagger\dagger\dagger}$
	ASPCR-DAG	$2.76 \pm 1.51^{\dagger}$	$2.76 \pm 1.51^{\dagger\dagger\dagger}$	2.76 ± 1.51
SF (8)	ABA-PC	8.15 ± 3.77	3.40 ± 2.81	12.8 ± 5.18
	OptABA-PC	$\mathbf{4.42 \pm 2.05}^{***}$	3.20 ± 1.81	$\mathbf{5.54 \pm 2.82}^{***}$
	MPC	$\mathbf{1.58 \pm 1.49}^{\dagger\dagger\dagger}$	$\mathbf{1.58 \pm 1.49}^{\dagger\dagger\dagger}$	$\mathbf{1.58 \pm 1.49}^{\dagger\dagger\dagger}$
	FGS	$\mathbf{2.50 \pm 1.96}^{\dagger\dagger\dagger}$	2.50 ± 1.96	$\mathbf{2.50 \pm 1.96}^{\dagger\dagger\dagger}$

Table 15: Average, best, and worst CPDAG F1 within each run of the matched primary experiments (mean ± standard deviation over seeds; higher is better). ABA-PC and OptABA-PC are scored over the saved classes satisfying their selected facts. MPC and FGS return one class when structurally valid; ASPCR-DAG's DAG supplies one class, so their average, best, and worst values coincide within a run. Stars mark the significant ABA-PC–OptABA-PC winner. Daggers compare an external method with OptABA-PC; bold daggers favour that method, unbold daggers favour OptABA-PC. One, two, and three symbols denote BH-adjusted $p < 0.05$, 0.01, and 0.001. Compatibility counts are in Table 18.

Dataset	Method	F1 avg	F1 best	F1 worst
Cancer (5)	ABA-PC	0.522 ± 0.214	0.646 ± 0.262	0.440 ± 0.222
	OptABA-PC	$\mathbf{0.576 \pm 0.259}^{*}$	0.576 ± 0.259	$\mathbf{0.576 \pm 0.259}^{**}$
	MPC	$\mathbf{0.661 \pm 0.202}^{\dagger}$	$\mathbf{0.661 \pm 0.202}^{\dagger}$	$\mathbf{0.661 \pm 0.202}^{\dagger}$
	FGS	$\mathbf{0.742 \pm 0.212}^{\dagger\dagger}$	$\mathbf{0.742 \pm 0.212}^{\dagger\dagger}$	$\mathbf{0.742 \pm 0.212}^{\dagger\dagger}$
	ASPCR-DAG	$0.490 \pm 0.123^{\dagger}$	$0.490 \pm 0.123^{\dagger}$	$0.490 \pm 0.123^{\dagger}$
Earthquake (5)	ABA-PC	0.991 ± 0.033	0.991 ± 0.030	0.989 ± 0.038
	OptABA-PC	0.991 ± 0.030	0.991 ± 0.030	0.991 ± 0.030
	MPC	0.991 ± 0.030	0.991 ± 0.030	0.991 ± 0.030
	FGS	$0.422 \pm 0.143^{\dagger\dagger\dagger}$	$0.422 \pm 0.143^{\dagger\dagger\dagger}$	$0.422 \pm 0.143^{\dagger\dagger\dagger}$
	ASPCR-DAG	$0.866 \pm 0.168^{\dagger\dagger\dagger}$	$0.866 \pm 0.168^{\dagger\dagger\dagger}$	$0.866 \pm 0.168^{\dagger\dagger\dagger}$
Survey (6)	ABA-PC	0.438 ± 0.088	$\mathbf{0.702 \pm 0.154}^{**}$	0.228 ± 0.120
	OptABA-PC	$\mathbf{0.597 \pm 0.127}^{***}$	0.602 ± 0.130	$\mathbf{0.593 \pm 0.126}^{***}$
	MPC	$\mathbf{0.815 \pm 0.088}^{\dagger\dagger\dagger}$	$\mathbf{0.815 \pm 0.088}^{\dagger\dagger\dagger}$	$\mathbf{0.815 \pm 0.088}^{\dagger\dagger\dagger}$
	FGS	$\mathbf{0.708 \pm 0.187}^{\dagger\dagger\dagger}$	$\mathbf{0.708 \pm 0.187}^{\dagger\dagger}$	$\mathbf{0.708 \pm 0.187}^{\dagger\dagger\dagger}$
	ASPCR-DAG	$0.496 \pm 0.154^{\dagger\dagger}$	$0.496 \pm 0.154^{\dagger\dagger}$	$0.496 \pm 0.154^{\dagger\dagger}$
Asia (8)	ABA-PC	0.224 ± 0.073	0.448 ± 0.083	0.111 ± 0.080
	OptABA-PC	$\mathbf{0.536 \pm 0.061}^{***}$	$\mathbf{0.540 \pm 0.063}^{***}$	$\mathbf{0.533 \pm 0.060}^{***}$
	MPC	$0.483 \pm 0.045^{\dagger\dagger\dagger}$	$0.483 \pm 0.045^{\dagger\dagger\dagger}$	$0.483 \pm 0.045^{\dagger\dagger\dagger}$
	FGS	0.561 ± 0.102	0.561 ± 0.102	0.561 ± 0.102
ER (5)	ABA-PC	0.562 ± 0.224	0.672 ± 0.230	0.475 ± 0.252
	OptABA-PC	$\mathbf{0.589 \pm 0.213}^{*}$	0.657 ± 0.227	$\mathbf{0.535 \pm 0.235}^{**}$
	MPC	$\mathbf{0.712 \pm 0.251}^{\dagger\dagger\dagger}$	0.712 ± 0.251	$\mathbf{0.712 \pm 0.251}^{\dagger\dagger\dagger}$
	FGS	$\mathbf{0.677 \pm 0.239}^{\dagger\dagger}$	0.677 ± 0.239	$\mathbf{0.677 \pm 0.239}^{\dagger\dagger\dagger}$
	ASPCR-DAG	$0.377 \pm 0.292^{\dagger\dagger\dagger}$	$0.377 \pm 0.292^{\dagger\dagger\dagger}$	$0.377 \pm 0.292^{\dagger\dagger}$
ER (8)	ABA-PC	0.478 ± 0.179	0.687 ± 0.149	0.309 ± 0.234
	OptABA-PC	$\mathbf{0.627 \pm 0.167}^{***}$	0.670 ± 0.172	$\mathbf{0.586 \pm 0.182}^{***}$
	MPC	$\mathbf{0.704 \pm 0.137}^{\dagger\dagger}$	0.704 ± 0.137	$\mathbf{0.704 \pm 0.137}^{\dagger\dagger\dagger}$
	FGS	$\mathbf{0.744 \pm 0.157}^{\dagger\dagger\dagger}$	$\mathbf{0.744 \pm 0.157}^{\dagger}$	$\mathbf{0.744 \pm 0.157}^{\dagger\dagger\dagger}$
SF (5)	ABA-PC	0.670 ± 0.176	0.813 ± 0.193	0.560 ± 0.221
	OptABA-PC	$\mathbf{0.737 \pm 0.148}^{**}$	0.851 ± 0.160	$\mathbf{0.646 \pm 0.201}^{**}$
	MPC	$\mathbf{0.874 \pm 0.173}^{\dagger\dagger\dagger}$	0.874 ± 0.173	$\mathbf{0.874 \pm 0.173}^{\dagger\dagger\dagger}$
	FGS	$\mathbf{0.854 \pm 0.173}^{\dagger\dagger\dagger}$	0.854 ± 0.173	$\mathbf{0.854 \pm 0.173}^{\dagger\dagger\dagger}$
	ASPCR-DAG	$0.448 \pm 0.303^{\dagger\dagger\dagger}$	$0.448 \pm 0.303^{\dagger\dagger\dagger}$	$0.448 \pm 0.303^{\dagger\dagger\dagger}$
SF (8)	ABA-PC	0.498 ± 0.154	0.739 ± 0.148	0.286 ± 0.185
	OptABA-PC	$\mathbf{0.629 \pm 0.129}^{***}$	0.690 ± 0.142	$\mathbf{0.581 \pm 0.144}^{***}$
	MPC	$\mathbf{0.774 \pm 0.163}^{\dagger\dagger\dagger}$	$\mathbf{0.774 \pm 0.163}^{\dagger\dagger}$	$\mathbf{0.774 \pm 0.163}^{\dagger\dagger\dagger}$
	FGS	0.683 ± 0.246	0.683 ± 0.246	$\mathbf{0.683 \pm 0.246}^{\dagger\dagger}$

Table 16: Average, best, and worst DAG SHD within each run of the matched primary experiments (mean ± standard deviation over seeds; lower is better). ABA-PC and OptABA-PC are scored over saved compatible DAGs; MPC and FGS over consistent extensions of each valid returned CPDAG; ASPCR-DAG on its single DAG, so its average, best, and worst coincide. Subscripts mark MPC's reduced samples: 38 on Survey and 49 on ER(8). Stars mark the significant ABA-PC–OptABA-PC winner. Daggers compare an external method with OptABA-PC; bold daggers favour that method, unbold daggers favour OptABA-PC. One, two, and three symbols denote BH-adjusted $p < 0.05$, 0.01, and 0.001. Compatibility counts are in Table 18.

Dataset	Method	SHD avg	SHD best	SHD worst
Cancer (5)	ABA-PC	2.98 ± 1.20	1.44 ± 1.18	4.18 ± 1.96
	OptABA-PC	$\mathbf{1.89 \pm 1.14}^{***}$	1.70 ± 1.15	$\mathbf{2.00 \pm 1.21}^{***}$
	MPC	1.70 ± 1.01	$\mathbf{0.86 \pm 0.61}^{\dagger\dagger\dagger}$	2.28 ± 1.37
	FGS	$\mathbf{1.11 \pm 0.94}^{\dagger\dagger\dagger}$	$\mathbf{0.90 \pm 0.81}^{\dagger\dagger\dagger}$	$\mathbf{1.30 \pm 1.18}^{\dagger\dagger}$
	ASPCR-DAG	$2.34 \pm 0.89^{\dagger}$	$2.34 \pm 0.89^{\dagger\dagger}$	2.34 ± 0.89
Earthquake (5)	ABA-PC	0.10 ± 0.34	0.08 ± 0.27	0.12 ± 0.48
	OptABA-PC	0.08 ± 0.27	0.08 ± 0.27	0.08 ± 0.27
	MPC	0.08 ± 0.27	0.08 ± 0.27	0.08 ± 0.27
	FGS	$4.41 \pm 1.15^{\dagger\dagger\dagger}$	$3.16 \pm 1.11^{\dagger\dagger\dagger}$	$5.32 \pm 1.38^{\dagger\dagger\dagger}$
	ASPCR-DAG	$1.06 \pm 1.20^{\dagger\dagger\dagger}$	$1.06 \pm 1.20^{\dagger\dagger\dagger}$	$1.06 \pm 1.20^{\dagger\dagger\dagger}$
Survey (6)	ABA-PC	6.73 ± 1.39	$\mathbf{2.56 \pm 1.47}^{*}$	10.3 ± 2.70
	OptABA-PC	$\mathbf{3.77 \pm 0.91}^{***}$	3.18 ± 0.92	$\mathbf{4.38 \pm 1.14}^{***}$
	MPC	$\mathbf{3.17 \pm 1.49}^{\dagger}_{38}$	$\mathbf{2.21 \pm 0.93}^{\dagger\dagger\dagger}_{38}$	$4.11 \pm 2.10_{38}$
	FGS	3.47 ± 1.28	$\mathbf{2.44 \pm 1.33}^{\dagger\dagger}$	4.52 ± 1.53
	ASPCR-DAG	$4.16 \pm 1.23^{\dagger}$	$4.16 \pm 1.23^{\dagger\dagger\dagger}$	4.16 ± 1.23
Asia (8)	ABA-PC	14.0 ± 2.81	8.02 ± 2.27	19.5 ± 3.45
	OptABA-PC	$\mathbf{4.51 \pm 0.58}^{***}$	$\mathbf{3.72 \pm 0.57}^{***}$	$\mathbf{5.02 \pm 0.82}^{***}$
	MPC	$\mathbf{3.92 \pm 0.25}^{\dagger\dagger\dagger}$	$\mathbf{2.82 \pm 0.44}^{\dagger\dagger\dagger}$	5.02 ± 0.14
	FGS	$6.01 \pm 1.70^{\dagger\dagger\dagger}$	$5.16 \pm 1.92^{\dagger\dagger\dagger}$	$6.76 \pm 1.61^{\dagger\dagger\dagger}$
ER (5)	ABA-PC	3.17 ± 1.73	1.34 ± 1.67	4.80 ± 2.26
	OptABA-PC	$\mathbf{2.49 \pm 1.35}^{***}$	1.04 ± 1.26	$\mathbf{3.76 \pm 1.78}^{***}$
	MPC	$\mathbf{1.23 \pm 0.94}^{\dagger\dagger\dagger}$	$\mathbf{0.46 \pm 0.93}^{\dagger}$	$\mathbf{1.94 \pm 1.22}^{\dagger\dagger\dagger}$
	FGS	$\mathbf{1.58 \pm 1.11}^{\dagger\dagger\dagger}$	0.72 ± 1.20	$\mathbf{2.38 \pm 1.34}^{\dagger\dagger\dagger}$
	ASPCR-DAG	$3.26 \pm 1.19^{\dagger\dagger}$	$3.26 \pm 1.19^{\dagger\dagger\dagger}$	3.26 ± 1.19
ER (8)	ABA-PC	7.99 ± 3.77	3.20 ± 2.49	12.5 ± 5.64
	OptABA-PC	$\mathbf{3.85 \pm 1.58}^{***}$	$\mathbf{2.14 \pm 1.67}^{**}$	$\mathbf{5.44 \pm 2.07}^{***}$
	MPC	$\mathbf{2.38 \pm 1.21}^{\dagger\dagger\dagger}_{49}$	$\mathbf{1.37 \pm 1.30}^{\dagger\dagger}_{49}$	$\mathbf{3.33 \pm 1.46}^{\dagger\dagger\dagger}_{49}$
	FGS	$\mathbf{2.07 \pm 1.41}^{\dagger\dagger\dagger}$	$\mathbf{0.96 \pm 1.54}^{\dagger\dagger}$	$\mathbf{3.08 \pm 1.58}^{\dagger\dagger\dagger}$
SF (5)	ABA-PC	3.07 ± 1.94	0.86 ± 1.47	4.88 ± 3.11
	OptABA-PC	$\mathbf{2.52 \pm 1.89}^{**}$	$\mathbf{0.56 \pm 1.28}^{*}$	$\mathbf{4.08 \pm 3.10}^{**}$
	MPC	$\mathbf{0.48 \pm 0.58}^{\dagger\dagger\dagger}$	$\mathbf{0.18 \pm 0.44}^{\dagger}$	$\mathbf{0.78 \pm 0.82}^{\dagger\dagger\dagger}$
	FGS	$\mathbf{0.69 \pm 0.85}^{\dagger\dagger\dagger}$	0.38 ± 0.81	$\mathbf{1.00 \pm 0.99}^{\dagger\dagger\dagger}$
	ASPCR-DAG	2.60 ± 1.26	$2.60 \pm 1.26^{\dagger\dagger\dagger}$	$\mathbf{2.60 \pm 1.26}^{\dagger}$
SF (8)	ABA-PC	8.14 ± 3.46	3.20 ± 2.69	12.7 ± 4.83
	OptABA-PC	$\mathbf{4.76 \pm 1.97}^{***}$	2.98 ± 1.73	$\mathbf{6.10 \pm 2.67}^{***}$
	MPC	$\mathbf{1.88 \pm 1.23}^{\dagger\dagger\dagger}$	$\mathbf{1.24 \pm 1.20}^{\dagger\dagger\dagger}$	$\mathbf{2.52 \pm 1.40}^{\dagger\dagger\dagger}$
	FGS	$\mathbf{2.82 \pm 1.88}^{\dagger\dagger\dagger}$	$\mathbf{2.08 \pm 1.77}^{\dagger}$	$\mathbf{3.56 \pm 2.10}^{\dagger\dagger\dagger}$

Table 17: Average, best, and worst DAG F1 within each run of the matched primary experiments (mean ± standard deviation over seeds; higher is better). ABA-PC and OptABA-PC are scored over saved compatible DAGs; MPC and FGS over consistent extensions of each valid returned CPDAG; ASPCR-DAG on its single DAG, so its average, best, and worst coincide. Subscripts mark MPC's reduced samples: 38 on Survey and 49 on ER(8). Stars mark the significant ABA-PC–OptABA-PC winner. Daggers compare an external method with OptABA-PC; bold daggers favour that method, unbold daggers favour OptABA-PC. One, two, and three symbols denote BH-adjusted $p < 0.05$, 0.01, and 0.001. Compatibility counts are in Table 18.

Dataset	Method	F1 avg	F1 best	F1 worst
Cancer (5)	ABA-PC	0.483 ± 0.223	0.688 ± 0.270	0.346 ± 0.264
	OptABA-PC	$\mathbf{0.563 \pm 0.262}^{*}$	0.616 ± 0.277	$\mathbf{0.531 \pm 0.281}^{***}$
	MPC	0.623 ± 0.222	$\mathbf{0.856 \pm 0.113}^{\dagger\dagger\dagger}$	0.462 ± 0.326
	FGS	$\mathbf{0.739 \pm 0.215}^{\dagger\dagger}$	$\mathbf{0.794 \pm 0.179}^{\dagger\dagger\dagger}$	$\mathbf{0.691 \pm 0.282}^{\dagger}$
	ASPCR-DAG	$0.462 \pm 0.225^{\dagger}$	$0.462 \pm 0.225^{\dagger\dagger}$	0.462 ± 0.225
Earthquake (5)	ABA-PC	0.989 ± 0.039	0.991 ± 0.030	0.985 ± 0.062
	OptABA-PC	0.991 ± 0.030	0.991 ± 0.030	0.991 ± 0.030
	MPC	0.991 ± 0.030	0.991 ± 0.030	0.991 ± 0.030
	FGS	$0.417 \pm 0.149^{\dagger\dagger\dagger}$	$0.652 \pm 0.145^{\dagger\dagger\dagger}$	$0.246 \pm 0.194^{\dagger\dagger\dagger}$
	ASPCR-DAG	$0.871 \pm 0.176^{\dagger\dagger\dagger}$	$0.871 \pm 0.176^{\dagger\dagger\dagger}$	$0.871 \pm 0.176^{\dagger\dagger\dagger}$
Survey (6)	ABA-PC	0.340 ± 0.103	$\mathbf{0.677 \pm 0.180}^{**}$	0.101 ± 0.135
	OptABA-PC	$\mathbf{0.482 \pm 0.140}^{***}$	0.586 ± 0.130	$\mathbf{0.373 \pm 0.204}^{***}$
	MPC	$0.528 \pm 0.242_{38}$	$\mathbf{0.720 \pm 0.134}_{38}^{\dagger\dagger\dagger}$	$0.342 \pm 0.367_{38}$
	FGS	0.465 ± 0.218	$\mathbf{0.660 \pm 0.236}^{\dagger}$	$0.268 \pm 0.270^{\dagger}$
	ASPCR-DAG	$0.385 \pm 0.220^{\dagger\dagger}$	$0.385 \pm 0.220^{\dagger\dagger\dagger}$	0.385 ± 0.220
Asia (8)	ABA-PC	0.310 ± 0.095	0.565 ± 0.105	0.109 ± 0.107
	OptABA-PC	$\mathbf{0.665 \pm 0.009}^{***}$	$\mathbf{0.754 \pm 0.045}^{***}$	$\mathbf{0.613 \pm 0.032}^{***}$
	MPC	$0.620 \pm 0.013^{\dagger\dagger\dagger}$	$\mathbf{0.786 \pm 0.037}^{\dagger\dagger\dagger}$	$0.454 \pm 0.014^{\dagger\dagger\dagger}$
	FGS	$0.610 \pm 0.105^{\dagger\dagger}$	$0.696 \pm 0.132^{\dagger\dagger}$	$0.533 \pm 0.094^{\dagger\dagger\dagger}$
ER (5)	ABA-PC	0.567 ± 0.215	0.817 ± 0.229	0.348 ± 0.264
	OptABA-PC	$\mathbf{0.620 \pm 0.188}^{**}$	0.847 ± 0.187	$\mathbf{0.417 \pm 0.248}^{**}$
	MPC	$\mathbf{0.768 \pm 0.177}^{\dagger\dagger\dagger}$	$\mathbf{0.924 \pm 0.170}^{\dagger}$	$\mathbf{0.623 \pm 0.242}^{\dagger\dagger\dagger}$
	FGS	$\mathbf{0.710 \pm 0.196}^{\dagger\dagger}$	0.886 ± 0.205	$\mathbf{0.547 \pm 0.255}^{\dagger\dagger}$
	ASPCR-DAG	$0.426 \pm 0.251^{\dagger\dagger\dagger}$	$0.426 \pm 0.251^{\dagger\dagger\dagger}$	0.426 ± 0.251
ER (8)	ABA-PC	0.496 ± 0.181	0.773 ± 0.167	0.261 ± 0.216
	OptABA-PC	$\mathbf{0.650 \pm 0.137}^{***}$	0.814 ± 0.154	$\mathbf{0.494 \pm 0.177}^{***}$
	MPC	$\mathbf{0.727 \pm 0.140}_{49}^{\dagger\dagger}$	$0.858 \pm 0.150_{49}$	$\mathbf{0.606 \pm 0.175}_{49}^{\dagger\dagger\dagger}$
	FGS	$\mathbf{0.774 \pm 0.139}^{\dagger\dagger\dagger}$	$\mathbf{0.916 \pm 0.146}^{\dagger\dagger\dagger}$	$\mathbf{0.643 \pm 0.175}^{\dagger\dagger\dagger}$
SF (5)	ABA-PC	0.594 ± 0.213	0.879 ± 0.201	0.370 ± 0.289
	OptABA-PC	$\mathbf{0.650 \pm 0.197}^{**}$	$\mathbf{0.915 \pm 0.181}^{*}$	$\mathbf{0.440 \pm 0.273}^{**}$
	MPC	$\mathbf{0.893 \pm 0.120}^{\dagger\dagger\dagger}$	$\mathbf{0.968 \pm 0.080}^{\dagger}$	$\mathbf{0.818 \pm 0.184}^{\dagger\dagger\dagger}$
	FGS	$\mathbf{0.861 \pm 0.143}^{\dagger\dagger\dagger}$	0.939 ± 0.127	$\mathbf{0.784 \pm 0.193}^{\dagger\dagger\dagger}$
	ASPCR-DAG	$0.448 \pm 0.283^{\dagger\dagger\dagger}$	$0.448 \pm 0.283^{\dagger\dagger\dagger}$	0.448 ± 0.283
SF (8)	ABA-PC	0.481 ± 0.166	0.769 ± 0.179	0.235 ± 0.197
	OptABA-PC	$\mathbf{0.601 \pm 0.146}^{***}$	0.733 ± 0.153	$\mathbf{0.499 \pm 0.182}^{***}$
	MPC	$\mathbf{0.773 \pm 0.143}^{\dagger\dagger\dagger}$	$\mathbf{0.871 \pm 0.130}^{\dagger\dagger\dagger}$	$\mathbf{0.674 \pm 0.181}^{\dagger\dagger\dagger}$
	FGS	0.664 ± 0.241	0.784 ± 0.212	0.542 ± 0.291

Table 18: Timing, structural output size, and CI compatibility for the matched primary experiments (mean ± standard deviation over 50 seeds for each reported method–dataset pair). Returned CPDAGs and DAGs are structural output/extension counts. CI-compatible counts additionally require all retained post-repair facts for ABA-PC/OptABA-PC or every recorded native test for MPC/ASPCR-DAG; compatibility is undefined for score-based FGS ('–'). ABA-PC and OptABA-PC times include construction, solving, and compatible-set evaluation and are directly paired. MPC/FGS time the learner call, while ASPCR-DAG reports its native end-to-end R run; their values and paired daggers describe implementations with different timing scopes and are not a controlled cross-family speed ranking. Stars identify the significant ABA-PC–OptABA-PC winner. Bold daggers favour the external method and unbold daggers favour OptABA-PC. One, two, and three symbols denote BH-adjusted $p < 0.05$, 0.01, and 0.001.

Dataset	Method	Time (s)	Returned CPDAGs	Returned DAGs	CI-compat. CPDAGs	CI-compat. DAGs
Cancer (5)	ABA-PC	1.51 ± 0.48	15.1 ± 28.8	31.9 ± 57.1	15.1 ± 28.8	31.9 ± 57.1
	OptABA-PC	$\mathbf{1.28 \pm 0.38}^{***}$	1.0 ± 0.0	1.5 ± 1.1	$\mathbf{1.0 \pm 0.0}^{***}$	$\mathbf{1.5 \pm 1.1}^{***}$
	MPC	$\mathbf{0.01 \pm 0.00}^{\dagger\dagger\dagger}$	1.0 ± 0.0	3.3 ± 1.6	0.0 ± 0.0	0.0 ± 0.0
	FGS	$\mathbf{0.08 \pm 0.09}^{\dagger\dagger\dagger}$	1.0 ± 0.0	1.8 ± 1.5	–	–
	ASPCR-DAG	$12.8 \pm 0.09^{\dagger\dagger\dagger}$	1.0 ± 0.0	1.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Earthquake (5)	ABA-PC	1.83 ± 0.83	1.0 ± 0.3	1.1 ± 0.5	1.0 ± 0.3	1.1 ± 0.5
	OptABA-PC	$\mathbf{1.68 \pm 0.81}^{***}$	1.0 ± 0.0	1.1 ± 0.2	1.0 ± 0.0	1.1 ± 0.2
	MPC	$\mathbf{0.01 \pm 0.00}^{\dagger\dagger\dagger}$	1.0 ± 0.0	1.1 ± 0.2	0.9 ± 0.2	1.0 ± 0.3
	FGS	$\mathbf{0.10 \pm 0.09}^{\dagger\dagger\dagger}$	1.0 ± 0.0	7.6 ± 6.2	–	–
	ASPCR-DAG	$12.7 \pm 0.08^{\dagger\dagger\dagger}$	1.0 ± 0.0	1.0 ± 0.0	0.4 ± 0.5	0.4 ± 0.5
Survey (6)	ABA-PC	15.7 ± 30.2	1278.5 ± 2807.2	4725.3 ± 9759.0	1278.5 ± 2807.2	4725.3 ± 9759.0
	OptABA-PC	2.94 ± 1.75	1.2 ± 0.7	2.9 ± 2.0	$\mathbf{1.2 \pm 0.7}^{***}$	$\mathbf{2.9 \pm 2.0}^{***}$
	MPC	$\mathbf{0.02 \pm 0.00}^{\dagger\dagger\dagger}$	0.8 ± 0.4	2.9 ± 2.4	0.0 ± 0.0	0.0 ± 0.0
	FGS	$\mathbf{0.08 \pm 0.09}^{\dagger\dagger\dagger}$	1.0 ± 0.0	4.2 ± 1.6	–	–
	ASPCR-DAG	$39.2 \pm 0.72^{\dagger\dagger\dagger}$	1.0 ± 0.0	1.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Asia (8)	ABA-PC	95.4 ± 48.8	10822.7 ± 7112.3	27191.5 ± 16181.0	10822.7 ± 7112.3	27191.5 ± 16181.0
	OptABA-PC	$\mathbf{66.5 \pm 84.9}^{*}$	1.2 ± 0.4	4.6 ± 3.8	$\mathbf{1.2 \pm 0.4}^{***}$	$\mathbf{4.6 \pm 3.8}^{***}$
	MPC	$\mathbf{0.04 \pm 0.01}^{\dagger\dagger\dagger}$	1.0 ± 0.0	4.8 ± 1.6	0.0 ± 0.0	0.0 ± 0.0
	FGS	$\mathbf{0.08 \pm 0.08}^{\dagger\dagger\dagger}$	1.0 ± 0.0	4.8 ± 1.6	–	–
ER (5)	ABA-PC	1.01 ± 0.15	8.2 ± 13.0	32.1 ± 61.1	8.2 ± 13.0	32.1 ± 61.1
	OptABA-PC	$\mathbf{0.90 \pm 0.08}^{***}$	3.0 ± 3.4	11.6 ± 15.6	$\mathbf{3.0 \pm 3.4}^{***}$	$\mathbf{11.6 \pm 15.6}^{**}$
	MPC	$\mathbf{0.01 \pm 0.00}^{\dagger\dagger\dagger}$	1.0 ± 0.0	3.3 ± 2.5	0.2 ± 0.4	0.6 ± 1.3
	FGS	$\mathbf{0.06 \pm 0.07}^{\dagger\dagger\dagger}$	1.0 ± 0.0	3.6 ± 2.6	–	–
	ASPCR-DAG	$15.5 \pm 0.23^{\dagger\dagger\dagger}$	1.0 ± 0.0	1.0 ± 0.0	0.1 ± 0.3	0.1 ± 0.3
ER (8)	ABA-PC	$\mathbf{48.0 \pm 54.3}^{***}$	3840.0 ± 5377.9	11539.7 ± 15313.7	3840.0 ± 5377.9	11539.7 ± 15313.7
	OptABA-PC	139.1 ± 139.4	4.7 ± 9.5	32.0 ± 64.1	$\mathbf{4.7 \pm 9.5}^{***}$	$\mathbf{32.0 \pm 64.1}^{***}$
	MPC	$\mathbf{0.02 \pm 0.01}^{\dagger\dagger\dagger}$	1.0 ± 0.1	5.2 ± 4.2	0.1 ± 0.2	0.2 ± 1.0
	FGS	$\mathbf{0.07 \pm 0.07}^{\dagger\dagger\dagger}$	1.0 ± 0.0	5.8 ± 4.5	–	–
SF (5)	ABA-PC	1.85 ± 2.03	88.4 ± 202.7	372.6 ± 856.2	88.4 ± 202.7	372.6 ± 856.2
	OptABA-PC	$\mathbf{1.76 \pm 2.05}^{***}$	85.1 ± 203.9	357.6 ± 861.0	$\mathbf{85.1 \pm 203.9}^{**}$	$\mathbf{357.6 \pm 861.0}^{**}$
	MPC	$\mathbf{0.01 \pm 0.00}^{\dagger\dagger\dagger}$	1.0 ± 0.0	1.7 ± 0.9	0.3 ± 0.5	0.5 ± 0.9
	FGS	$\mathbf{0.06 \pm 0.08}^{\dagger\dagger\dagger}$	1.0 ± 0.0	1.7 ± 1.1	–	–
	ASPCR-DAG	$15.4 \pm 0.25^{\dagger\dagger\dagger}$	1.0 ± 0.0	1.0 ± 0.0	0.1 ± 0.3	0.1 ± 0.3
SF (8)	ABA-PC	$\mathbf{42.3 \pm 53.4}^{**}$	3108.1 ± 4287.4	8716.5 ± 11926.2	3108.1 ± 4287.4	8716.5 ± 11926.2
	OptABA-PC	99.1 ± 122.0	170.3 ± 1023.2	969.8 ± 6017.6	$\mathbf{170.3 \pm 1023.2}^{***}$	$\mathbf{969.8 \pm 6017.6}^{***}$
	MPC	$\mathbf{0.02 \pm 0.01}^{\dagger\dagger\dagger}$	1.0 ± 0.0	2.8 ± 1.4	0.0 ± 0.2	0.1 ± 0.5
	FGS	$\mathbf{0.06 \pm 0.07}^{\dagger\dagger\dagger}$	1.0 ± 0.0	3.3 ± 1.8	–	–