Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Bipolar Logic Networks mine condition-dependent mechanism candidates with fully auditable configurations

Shidong Li

20220054@nuc.edu.cn

North University of China

Article

Keywords:

Posted Date: November 5th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11239564/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Abstract

In high-stakes domains, a discovered relationship is useful only if its holding conditions can be stated explicitly and inspected by an expert. Causal-discovery algorithms return a graph of adjacent variables, but none states whether a parent is necessary, sufficient, or merely contributory, and post hoc explanations need not match the model's actual computation. We present Bipolar Logic Networks (BLN), a neural-symbolic architecture with two interpretable paths—a sufficient disjunctive-normal-form path and a necessary conjunctive-normal-form path—fused by a learned gate; every reported configuration is a trained minterm or clause exercised in the predictions it makes. Across the Sachs, ALARM, and CHILD networks, 162 synthetic datasets, and a four-environment intervention experiment, BLN recovered nearly the entire ground-truth skeleton (recall 0.95, 0.96, 1.00), and on Sachs 12 of 18 recovered true edges were missed by both PC and GES. Conjunctions validated as wholes on held-out data reached positive purity several-fold above base rate, and non-empty stability separated a stable core (Jaccard 0.79–0.88 over 12–13 conditions) from condition-specific targets. Within the boundary that outputs are mechanism candidates, not identified causal effects, BLN complements structure learners by supplying roles, validated conditions, and stability that graphs lack.

Introduction

In high-stakes decision settings—clinical diagnosis, public health, credit, safety—a decision must be more than correct. It must be *defensible*: a reviewer must be able to inspect the conditions that produced it, assess whether they are complete, and trace responsibility when those conditions fail. Regulatory regimes from the GDPR to the U.S. Executive Order on artificial intelligence codify this expectation [1, 2], and the causal discovery and interpretable machine learning communities have each responded to it. Yet the two responses remain separated by a conceptual gap.

Causal discovery algorithms identify *whether* a dependence exists under assumptions, but not the *logical role* a parent plays. A graph recovered by PC [3], GES [4], NOTEARS [5], or DAGMA [6] records that node A is adjacent to node B. It does not record whether A is necessary for B, sufficient for B under particular conditions, one of several INUS conditions [7], or merely contributory. An expert confronting the edge A → B in a clinical model cannot ask the graph: "Under what configuration of A—and of the other parents—is B triggered, and what would block it?" The edge exists, but the condition is missing.

Necessary condition analysis [8] and configurational methods such as qualitative comparative analysis [9] address roles directly, but as separate analytical procedures applied after modelling. The configurations they report are not necessarily the configurations an operational model uses; when the model is subsequently retrained, audited, or contested, the NCA/QCA layer and the model layer can silently diverge. The auditability is procedural rather than architectural, and it does not survive contact with the system as deployed.

Post hoc explanation methods exhibit a parallel problem in reverse. SHAP and LIME attach explanations to an already-trained model, but the explanations are not guaranteed to correspond to the internal

conditions the model relies on; they are local approximations that can differ between methods, between perturbations, and between runs. A reviewer given a LIME explanation [10] is reading a description of the model produced by another model, with no guarantee that the two will agree on a new case. Model cards and documentation address governance at the organisational level but remain static artefacts that can drift out of alignment with a changed system. Auditability attached after the fact is, at best, only as stable as the attachment.

Table 1 summarises the resulting capability landscape. No existing class of method simultaneously (i) distinguishes necessary from sufficient roles, (ii) places configurations inside the trained model so that they are computationally exercised rather than attached, (iii) validates each conjunction as a whole on held-out data, and (iv) reports cross-environment stability with an explicit detection denominator. Each existing class occupies one or two columns; the conjunction of the four columns is empty.

Table 1
Capability/problem matrix: what each class of method provides.

Method class	Outputs edges / graph	Distinguishes necessary vs. sufficient roles	Configurations are part of the trained model (not post hoc)	Whole conjunction validated on held-out data	Cross-environment stability reported
PC	yes	no	n/a	no	no
GES	yes	no	n/a	no	no
NOTEARS / DAGMA	yes	no	n/a	no	no
NCA / QCA	partial	**yes**	no (separate procedure)	no	partial
SHAP / LIME (post hoc)	no	partial	no (attached)	no	no
BLN (this work)	**yes**	**yes**	**yes**	**yes**	**yes**

Checks describe the method class as used in standard practice; “n/a” indicates that the method outputs a graph without role semantics rather than configurations. Sources for the standard-practice descriptions are given in Methods and References.

What is missing is an architecture in which the logical configuration is part of the model’s own structure —learned from data, exercised on every prediction, and inspectable by construction—rather than a graph augmented with role annotations or a prediction augmented with an explanation.

We develop such an architecture. A *Bipolar Logic Network* is a neural network built from two interpretable logical paths (Fig. 1). A sufficient path implements disjunctive normal form: conjunctive minterms (AND of literals) combined by a differentiable OR, so that any one activated minterm can trigger the outcome.

A necessary path implements conjunctive normal form: disjunctive clauses (OR of literals) combined by an AND, so that every required clause must be satisfied. A learned gate fuses the two paths. Because the minterms and clauses are the network's actual computational units, the configurations an expert audits are exactly the configurations the network uses; there is no explanatory layer that could disagree with the model.

This design makes a specific methodological bet: that logical necessity and sufficiency should be *architectural*, built into what is trained, not attached afterward. We test that bet on three benchmark networks with established ground truth—Sachs, ALARM, and CHILD—on 162 synthetic datasets spanning linear and nonlinear mechanisms, Erdős–Rényi and scale-free [11] structures, and dimensionality from 10 to 50 nodes, and on a four-environment soft-intervention experiment. We evaluate skeleton recall, precision, and structural Hamming distance against both classical and continuous structure learners, we validate each reported conjunction as a whole on held-out data, we recompute configuration stability on a non-empty basis, and we control the false discovery rate of the auxiliary screening layer.

The findings map the two sides of the architectural trade-off. On one side, the trained network recovers almost every true edge—consistent with benchmark comparisons of structure-learning methods [12]—genuine conjunction shrinks the configuration set and raises its held-out purity, and the 14-condition record separates a stable signalling core from condition-specific targets. On the other side, the candidate set is broad rather than sparse—precision is low, and on clean continuous data the graph produced by purpose-built continuous learners is markedly closer to ground truth. Read together, the two sides define a complementary division of labour: classical structure learners provide the sparse graph, and BLN provides the roles and validated conditions attached to its edges—auditable *mechanism candidates* that an expert can confirm or reject.

The contributions are:

1. To our knowledge, the first architecture in which logical necessity and sufficiency are part of the trained model itself: sufficient DNF and necessary CNF paths constitute the trained network, configurations are learned and computationally exercised, and every prediction is traceable to the specific minterms and clauses that produced it.
2. A network-grounded extraction and validation protocol: edges and roles are exported directly from trained logical paths, and every conjunctive minterm is validated as a whole on held-out data, not assembled from individually significant variables.
3. A non-empty definition of configuration stability over the 14 Sachs experimental conditions, in which two empty detections are not counted as agreement, alongside Benjamini–Hochberg control of the auxiliary screening layer and category-native baselines for categorical networks.
4. An empirical record reported without selection—including adverse synthetic and pooling results, and an early version of this work [13]—so that the boundary between BLN's candidate-generation role and the structure-accuracy role of classical learners is independently reproducible.

Results

High-recall trained-network extraction

For every target node we trained a complete BLN—a sufficient DNF path, a necessary CNF path, and the learned fusion gate—on the training split, exported minterms and clauses directly from the trained network, and scored the resulting skeleton against the consensus network. Two versions are reported throughout: the *raw* network, which retains every edge present in an exported minterm or clause, and the *predictively qualified* network, which additionally requires that removing the path's contribution degrades held-out prediction. All training was deterministic, single-threaded, CPU-only, with recorded seeds.

Table 2 reports the headline skeleton metrics. The raw trained network recovered the full skeleton of Sachs (recall 1.00) and CHILD (1.00) and nearly all of ALARM (1.00 raw, 0.96 qualified); the candidate set is correspondingly broad: raw skeleton precision ranged from 0.07 (ALARM) to 0.35 (Sachs), and raw SHD exceeded the number of true edges on all three networks. Predictive qualification removed a small number of edges—improving SHD on Sachs from 36 to 32 and on ALARM from 620 to 563—but did not change the qualitative picture: BLN is a high-recall extractor whose candidate set is built for expert review. We report the qualified network as the headline because its edges carry predictive evidence; both raw and qualified SHD values are shown in Table 2.

Table 2
Skeleton metrics of edges extracted from trained BLN networks.

Network	True edges	Raw recall	Raw precision	Raw SHD	Qualified recall	Qualified precision	Qualified SHD
Sachs (n = 853)	19	1.00	0.35	36	0.95	0.37	32
ALARM (n = 5,000)	46	1.00	0.07	620	0.96	0.07	563
CHILD (n = 5,000)	25	1.00	0.13	165	1.00	0.13	163

Restricting each node to its locally strongest configurations—reporting only the top three, five, or eight mechanisms per target—traded recall for precision as expected and gives reviewers a graded entry point into the candidate set (Table 3). On ALARM, a top-three restriction cut the estimated edge count from 605 to 256 and reduced SHD from 563 to 260 while preserving 0.46 recall; on Sachs the same restriction gave the best SHD of any variant (26) at recall 0.79. Cap-based local reporting is therefore an available dial for expert review—show a reviewer the strongest candidates first—but it is an information filter, not evidence that the discarded configurations are false.

Table 3
Skeleton metrics under local-mechanism caps (qualified network).

Network	Cap	Recall	Precision	SHD	Estimated edges
Sachs	3	0.79	0.41	26	37
Sachs	8	0.90	0.36	32	47
ALARM	3	0.46	0.08	260	256
ALARM	8	0.50	0.09	269	269
CHILD	3	0.72	0.14	117	128
CHILD	8	0.72	0.14	118	129

True conjunction and held-out validation

The sufficient path's minterms are conjunctive by construction: a minterm activates only when *all* of its literals hold simultaneously. For each qualified minterm containing at least two literals, we evaluated the conjunction as a whole on the stratified 30% held-out split—recording how often the conjunction triggered, the positive-class purity when it triggered, and the lift over the held-out base rate—and compared it against the configuration count that would result from reporting individually significant variables side by side.

Across the three trained networks the sufficient paths contained 17 (Sachs), 99 (ALARM), and 72 (CHILD) active minterms, of which the great majority on ALARM (97) and CHILD (70) were genuine multi-literal conjunctions rather than single literals. Restricting to conjunctions that fired on at least 30 held-out cases and that exceeded the held-out base rate left 10 (Sachs), 58 (ALARM), and 50 (CHILD) validated sufficient minterms; their median positive-class purity exceeded the base rate by 0.43, 0.67, and 0.49 in absolute terms, corresponding to median purity ratios of 1.9, 4.6, and 3.3. The conjunction requirement is therefore not merely structural—it selects a minority of minterms that jointly reproduce the positive class out of sample at several times the base rate, while the minterms that fail either trigger criterion are excluded rather than reported as sufficient.

Necessary clauses were evaluated on the complementary criterion: the rate at which each clause holds *among held-out positive cases*, together with the clause's base rate among all held-out cases. The trained networks produced 85 (Sachs), 350 (ALARM), and 241 (CHILD) active clauses; their median positive-class satisfaction was 0.87, 0.98, and 0.90. Clauses with a high positive satisfaction but a similarly high base rate—common on ALARM (median base rate 0.76)—are reported as weak necessity (high coverage, low discriminating power), not as strong necessary conditions, because coverage and discrimination are recorded separately for every clause.

Edges BLN recovers that sparse learners miss

The low precision of the high-recall candidate set raises a sharper question than SHD: does the network recover *true* edges that a sparse-graph workflow would miss entirely? On Sachs, where every edge can be checked against the 19-pair consensus skeleton, we compared the qualified BLN skeleton with PC (Fisher-z) and GES (Gaussian score), run on the same samples [14]. PC and GES each recovered 7 of the 19 consensus pairs; BLN recovered 18. Of BLN's 18 true pairs, **12 were missed by both PC and GES**—the Mek–Erk cascade link, PIP3–Akt, PIP2–Plcg, the PKC links to Raf, Mek, Plcg, and PIP2, the PKA links to Raf, Mek, PIP3, and the stress-target links P38–PKA and Jnk–PKA (full edge-by-edge list, with the minterm or clause and held-out evidence for each, in Supplementary Table 8). Each of the 12 is an edge of the Sachs consensus network and appears in a qualified minterm or clause with its held-out evidence attached. The reverse direction is also informative: exactly one consensus pair, Plcg–PIP3, was recovered by PC and GES but dropped by predictive qualification, so the two pipelines miss different single edges. Six pairs were recovered by all three methods—including the stress-kinase links P38–PKC and Jnk–PKC, which the directed arrows of the PC/GES CPDAGs do capture.

BLN still proposes many extra false pairs, and on clean continuous data PC/GES are the appropriate tools for obtaining a sparse graph. The broad candidate set, however, contains true edges—with roles and validated conditions—that a sparse-only workflow does not surface. An expert can filter BLN's candidates against the PC/GES graph while inspecting beyond the sparse result.

A second validity question is whether the trained models predict out of sample at all, since configurations extracted from a non-predictive model would not be its real computation. Reconstructing each frozen trained model and scoring it on its recorded held-out split gave mean held-out ROC-AUC / PR-AUC of 0.74 / 0.75 on Sachs [15, 20], 0.88 / 0.81 on ALARM (37 nodes), and 0.83 / 0.73 on CHILD (20 nodes) (per-target values in Supplementary Table 9). Together with the conjunction validation, this supports the claim that the reported configurations are units the trained model actually uses rather than post hoc labels.

Non-empty stability over experimental conditions

Sachs comprises 14 experimental conditions in addition to the combined observational subset. We trained the full extraction pipeline independently within each condition and recomputed stability on a non-empty basis: for each target, the number of conditions in which a condition was actually detected is reported first, and the mean pairwise Jaccard index is then computed only over pairs of conditions in which at least one configuration was detected. Two empty detections contribute no agreement—a departure from the convention under which two empty sets are counted as perfectly stable.

Table 4
Non-empty stability across the 14 Sachs conditions.

Target	Conditions detected / 14	Non-empty pairs	Mean pairwise Jaccard	Note
Raf	13	78	0.80	stable core
Mek	13	78	0.79	stable core
Erk	10	45	1.00	identical across detections
PIP2	12	66	0.88	stable core
PIP3	12	66	0.72	stable core
Akt	9	36	0.90	stable across detections
Plcg	9	36	0.50	moderate
PKA	6	15	0.64	moderate
PKC	4	6	0.62	condition-specific
Jnk	3	3	0.49	condition-specific
P38	1	0	not defined	only one detection; no supported statement

Three consequences follow. First, the previously reported headline necessity stability of 1.00 was an artefact of conditions in which the target produced no detection and two empty sets were scored as agreement; under the non-empty definition the strongest targets stabilise across 0.79–0.88 over 12–13 detected conditions, with Erk reaching 1.00 over its 10 detections. Second, targets that are genuinely condition-specific (PKC, Jnk) are now distinguishable from targets with a stable core (Raf, Mek, PIP2), which the empty-equal-stable convention obscured. Third, a target with only one detection—P38, detected in a single condition—is reported as having no statistically supported stability statement rather than as “stable”; absence of evidence is not converted into evidence of stability. The detection count and the Jaccard value are always presented together, so a high Jaccard on very few detected conditions cannot be read as robustness across all 14.

Complementary use with structure learners

Categorical networks require categorical tests. On ALARM and CHILD we ran PC with chi-square and G-square conditional-independence tests and GES with the BDeu categorical score [16]—the category-native comparisons—while retaining NOTEARS-linear and DAGMA-linear as continuous methods applied to integer category codes, with their encoding disadvantage labelled explicitly. On Sachs all variables are continuous, so PC (Fisher-z), GES (Gaussian score), NOTEARS, and DAGMA are the native comparisons. Every method received the same samples and a comparable budget.

Table 5
Skeleton SHD by method (lower is better).

Method	Sachs	ALARM	CHILD
PC (native test)	13 (Fisher-z)	7 (chi2)	0 (chi2)
GES (native score)	13 (Gaussian)	3 (BDeu)	0 (BDeu)
NOTEARS-linear	17	47 (codes; disadvantaged)	24 (codes; disadvantaged)
DAGMA-linear	15	34 (codes; disadvantaged)	14 (codes; disadvantaged)
DirectLiNGAM	14	—	—
BLN (qualified)	**32**	**563**	**163**

The two metrics locate the two methods at their respective strengths. Measured on skeleton *recall*, BLN ranks first on all three networks; measured on SHD, the sparse-graph methods rank ahead, by a wide margin on the categorical networks: category-native GES and PC reconstructed the CHILD skeleton exactly (SHD 0) and reached SHD 3 (GES) and 7 (PC) on ALARM, where BLN reached 563, because every extra candidate edge is penalised once in SHD and the categorical PC/GES algorithms are purpose-built for exactly this encoding. On Sachs the margin is smaller (32 versus 13–17). These numbers establish a division of labour: PC/GES answer “which variables are connected, sparsely,” and BLN answers, for the edges it recovers, “what is each parent’s role and under which validated conjunction does the edge fire.” An expert session on a categorical network therefore runs both—category-native PC or GES for the graph, BLN for roles and conditions—and filters BLN’s extra candidates against the sparer graph.

Multiplicity control

The auxiliary screening layer performs one test per candidate feature [17] (and per category indicator on categorical data). We applied Benjamini–Hochberg correction at a target false discovery rate of 0.05 and report significant counts before and after correction. On Sachs, 12 of 110 feature-level tests were nominally significant and 11 survived correction—the multiplicity burden is modest at 11 candidates. On ALARM the burden was substantial: 1,842 of 10,920 tests were nominally significant under the primary test (3,824 under the G-square variant), of which 1,540 (3,474) survived BH correction, removing 302 nominal discoveries; on CHILD, 1,132 of 3,540 were nominal and 1,010 survived (122 removed). FDR correction therefore trims the low-precision tail but, on dense categorical screens, leaves the great majority of nominal findings in place; it does not by itself produce a sparse structure. Screening is auxiliary in our protocol—the trained network is the source of edges—so corrected p-values gate which candidates reach training rather than serving as the evidence for an edge.

Scope boundary

On clean continuous data generated by an identifiable structural equation model [18], purpose-built continuous learners are the appropriate structure tools; the full synthetic benchmark, including per-dataset SHD for all eight methods across all 162 datasets, is reported in Supplementary Tables 4–6. Details of the four-environment soft-intervention experiment are likewise reported in full in the Supplementary Information.

Discussion

What the architecture delivers

The central design claim was that logical necessity and sufficiency should be built into the trained model, not attached afterward. The experiments identify what that choice buys. It buys configurations that are *exercised*: every reported sufficient minterm and necessary clause is a unit of the trained network, active on a subset of predictions, and removable in a controlled qualification test—so an audit trail points to a computation that actually occurred, not to a post hoc reconstruction. It buys *conjunction by default*: because a minterm activates only under joint literals, the failure mode of assembling individually significant variables into an apparent multi-factor rule is structurally unavailable. And it buys *role separation*: the sufficient and necessary paths can be inspected, validated, and capped independently, which a single adjacency matrix does not support.

Trade-offs of the architecture

The same experiments make the cost of the design explicit: end-to-end auditable configurations versus sparse-graph accuracy. On the real benchmark networks BLN's SHD is higher than that of purpose-built learners (Table 5), because the logical architecture that makes configurations inspectable also keeps the candidate set broad, and the qualification, conjunction, cap, and FDR mechanisms sharpen but do not sparsify it; on clean continuous data from an identifiable structural equation model the boundary likewise favours purpose-built continuous learners (Supplementary Tables 4–6). This is a deliberate trade-off between two design objectives, not a failure of either: the architecture was specified to prioritise recovering candidate mechanisms with their roles and conditions, and the graph-accuracy objective is the one classical structure learners already serve well. The practical consequence is complementary use by construction—the sparse graph from PC/GES, the annotated roles and validated conjunctions from BLN.

Network-grounded evidence versus statistical screening

A methodological consequence concerns where the evidence for an edge lives. In a screening-first pipeline, a reported edge is a rejected null; if the test is misspecified for the data type—or if individually significant variables are later presented as a conjunction—the reported structure inherits the misspecification. In the network-grounded protocol the edge is a unit that exists in a trained model, predicts out of sample, survives a contribution test, and is validated as a whole on held-out data; permutation-based and categorical tests remain in the pipeline only to gate which candidates are

trained. This is strictly more evidence per reported edge, and it removes the specific failure modes that motivated the revision. Predictive real edges can still be false against the consensus graph—the breadth of the candidate set shows as much.

Stability is not uniformity

The stability analysis makes a point with broader applicability. A configuration that is detected in only a fraction of environments can be made to look perfectly stable by the convention that scores two absences as agreement. Reporting the detection denominator alongside the Jaccard value separates three states that should never be conflated: a configuration present and consistent across many environments (a candidate stable core), present in a few environments and varying across them (condition-specific), and essentially absent (no supported statement). Under the non-empty definition the Sachs cascade shows exactly this gradient, from Raf and Mek at 0.79–0.80 across thirteen conditions to P38 with a single detection and no defined stability. Uniformity of a rule across environments is a strong claim that requires the rule to be *present* in those environments; we provide the counts that let a reader verify it.

Limitations

Five limitations bound the conclusions. First, observational configurations express logical semantics over statistical associations; they are mechanism candidates, not identified causal effects, and none of the reported roles should be read as a causal effect estimate. The controlled interventional evidence available for Sachs is what justifies causal language in that domain, and it does not transfer automatically to observational settings. Second, BLN is evaluated on three benchmark networks and a structured synthetic family; these cover common continuous and categorical regimes but not time-series, latent-confounded, or large-scale settings, on which we report nothing. Third, the precision gap is not closed: predictive qualification, conjunction, caps, and FDR correction reduce the surplus but leave BLN well below purpose-built learners in SHD, so an expert review step is part of the method. Fourth, the stability analysis is conditional on the Sachs experimental design—14 environments with differing intervention targets and sample sizes—and aggregate Jaccard values should be read together with the per-condition detection counts rather than as a single robustness number. Fifth, naive pooling of soft-intervention environments did not help; whether environments can help under an invariant or intervention-aware objective is untested here (full details in the Supplementary Information).

Future work

Four directions follow. Scaling the architecture toward sparse, calibrated structure—through stronger sparsity priors or joint training against a classical skeleton—might narrow part of the precision gap without sacrificing auditability. Naive pooling of environments did not help under the current procedure; coupling BLN to invariant or intervention-aware objectives is the direction that would turn the multi-environment setting into a source of identification. Extending evaluation to time-resolved and latent-confounded data would test whether the configuration semantics survive temporal and unobserved confounding. And formalising the human-review interface—the order in which capped candidates are

presented, contested, and confirmed—would turn the architectural audit trail into a practical governance workflow.

Conclusion

Causal discovery tells us whether variables are connected, producing the sparse graph that high-stakes workflows depend on; post hoc explanation tells us, approximately, why a model behaved as it did. Bipolar Logic Networks occupy a distinct and, to our knowledge, previously unfilled position: logical necessity and sufficiency are part of the trained model, exercised on every prediction, and inspectable by construction. On established benchmarks the network recovered almost every true edge—0.95, 0.96, and 1.00 skeleton recall—genuine conjunctions were validated as wholes on held-out data at several times the base rate, and the non-empty stability record separated a stable signalling core from condition-specific targets. That auditable layer is what sparse structure learners do not provide and what high-stakes expert review requires; the complementary workflow—PC, GES, NOTEARS, or DAGMA for the graph, BLN for the roles and conditions attached to its edges—uses each instrument for what it establishes. Within the mechanism-candidate boundary, BLN supplies, alongside training, configurations an expert can inspect, debate, and test, with a receipt showing exactly which logical units produced them.

Methods

Bipolar logical architecture

Each target node is modelled by a bipolar network with two interpretable paths and a learned fusion gate. The sufficient path is a differentiable implementation of disjunctive normal form. Input literals $l_1, \ldots, l_m$—signed activations encoding "feature j is high" or "feature j is low"—are combined into conjunctive minterms by a soft AND (minimum aggregation with a learned temperature), and the minterms are combined into a sufficient activation f_{suff} by a soft OR (a softmax-weighted sum), so that any single activated minterm drives the path:

$$f_{\text{suff}} = \text{SoftOR}_k \left(\text{SoftAND}_{j \in M_k} \left(l_j \right) \right).$$

The necessary path is a differentiable implementation of conjunctive normal form. Literals are combined into disjunctive clauses by a soft OR, and the clauses are combined into a necessary activation f_{nec} by a soft AND, so that every required clause must hold:

$$f_{\text{nec}} = \text{SoftAND}_k \left(\text{SoftOR}_{j \in C_k} \left(l_j \right) \right).$$

A fusion layer combines the paths with a learned, normalised gate α:

$$h = \alpha_s f_{\text{suff}} + \alpha_n f_{\text{nec}}, \quad \hat{y} = \sigma\left(h\right),$$

where α_s, α_n are produced by a softmax so that $\alpha_s + \alpha_n = 1$. Training jointly optimises paths and gate by binary cross-entropy together with margin, ordering, and anti-collapse regularisers that prevent path degeneracy and preserve configurational diversity. All operators are differentiable; thresholds are fixed at 0.5 and are not tuned per dataset.

Network-grounded extraction

After training, each minterm and clause is read back from the trained path parameters: the literals of a minterm identify the conjunction, and the literals of a clause identify the disjunction. An edge $i \to j$ is reported when feature i appears as a literal in at least one non-degenerate minterm or clause of the trained network for target j, and the edge is tagged sufficient, necessary, or both according to the path in which it appears. The raw extracted network retains every such edge. The predictively qualified network additionally requires that the path in which the edge participates contribute to held-out performance, assessed by removing the path's contribution and measuring the degradation of the held-out log-likelihood and classification; edges whose paths contribute nothing are reported in the raw metrics but excluded from the qualified headline. No permutation statistic is the sole source of an edge: permutation and categorical tests serve only to screen which candidate literals reach training.

True-conjunction held-out validation

Each dataset is split 70/30 into training and held-out sets using a fixed seed (13010) with stratification on the binary target. Only the training split is used to fit the network. For every qualified sufficient minterm with at least two literals, the conjunction is evaluated as a whole on the held-out split: we record the number and fraction of held-out cases in which all literals hold simultaneously, the positive-class rate among those cases, the held-out positive base rate, and their ratio (lift), with a Wilson interval for the trigger rate. A minterm is reported as a sufficient configuration only when the conjunction jointly triggers and its held-out positive rate exceeds the base rate by a recorded margin. Necessary clauses are evaluated by their satisfaction rate among held-out positives, together with satisfaction among negatives, so that coverage and discrimination are reported separately. Individually significant variables are never concatenated into a reported minterm.

Non-empty stability

For each Sachs target, the pipeline is applied independently within each of the 14 experimental conditions, producing a configuration set $R_c(T)$ per condition. The number of conditions with a non-empty detection, K_{det}, is reported. Stability is the mean pairwise Jaccard index over pairs in which at least one set is non-empty:

$$\text{Stability}\,(T) = \frac{1}{N_{\text{pairs}}} \sum_{\substack{a<b \\ R_a \cup\, R_b \neq\, \varnothing}} \frac{|R_a \cap\, R_b|}{|R_a \cup\, R_b|},$$

where N_{pairs} counts pairs with a non-empty union. Pairs of two empty sets are excluded rather than scored as one, and pairs with one empty and one non-empty set enter the average as zero overlap. Targets with zero detections are labelled as having no significant condition; targets with exactly one detection have no defined pair statistic. Values are computed separately for sufficient and necessary configurations.

Multiplicity control

Each auxiliary screening test is indexed by (target, candidate feature), and additionally by category level on categorical data, where features are supplied as category indicators. Raw p-values are adjusted within target by the Benjamini–Hochberg procedure at $q = 0.05$; both the nominal and the corrected significant counts are reported. Categorical screens use chi-square and G-square tests of independence; the continuous screen uses the permutation-based effect-size test. A candidate whose corrected p-value exceeds q does not enter training. Correction is performed before network fitting so that the multiplicity adjustment cannot be influenced by downstream results.

Datasets and ground truth

Sachs. Observational measurements of 11 phosphorylated proteins and phospholipids under combined anti-CD3/CD28 stimulation [14] (n = 853), with the consensus signalling network (19 skeleton edges, 20 directed edges) as ground truth. The full dataset comprises 14 experimental conditions used for the stratified stability analysis; condition-level sample sizes and intervention targets are read from the distributed metadata and held fixed.

ALARM. A 37-node, 46-edge Bayesian network for anaesthesiology alarm handling [19]; 5,000 samples with categorical variables of cardinality 2–4, and the network topology as ground truth.

CHILD. A 20-node, 25-edge Bayesian network for neonatal disease diagnosis [21]; 5,000 samples with categorical variables of cardinality 2–6, and the network topology as ground truth.

Synthetic. 162 datasets drawn from Erdős–Rényi and scale-free graph structures crossed with linear, linear non-Gaussian, and nonlinear structural mechanisms at 10, 20, and 50 dimensions (n = 500 per dataset), generated under fixed seeds with the true adjacency matrix retained.

Multi-environment. One observational environment (n = 2,000) and three soft-intervention environments over the same 10-node structure, with the intervention targets and mechanism shifts recorded in metadata; pooled analyses down-sample to a matched sample size.

Baselines

PC is run with the data-native conditional-independence test—Fisher-z on continuous data, chi-square and G-square on categorical data—at $\alpha = 0.05$ [3]. GES is run with the Gaussian BIC score on continuous data and the BDeu categorical score on categorical data. DirectLiNGAM [11] is additionally run on the continuous Sachs data. NOTEARS-linear [5] and DAGMA-linear [6] are run on continuous data and, on categorical datasets, on integer category codes with the encoding disadvantage labelled. All methods use the same samples, the same 70/30 splits where applicable, a fixed seed (13010), and a comparable computational budget. Continuous methods on categorical data are never compared against category-native methods without the label.

Metrics

Skeleton metrics treat an edge as present if either direction is reported. With G the recovered skeleton and G^* the true skeleton, recall is $|G \cap G^*| / |G^*|$, precision is $|G \cap G^*| / |G|$, and the skeleton SHD is $|G \triangle G^*|$ (false positives plus false negatives). Directed SHD additionally counts reversed arcs once. Edges exported from the network are matched to ground truth by node identity; no remapping that could inflate agreement is performed.

Computational environment

All experiments were executed on CPU using Python 3.13.12 and PyTorch 2.13.0; no GPU or special hardware was required.

Declarations

Ethics approval and consent to participate

This study used only publicly available, fully de-identified benchmark datasets (Sachs, ALARM, CHILD, and synthetic data). No new human-participants data were collected and no identifiable individuals are involved; therefore no ethical approval was required, and consent to participate and consent for publication are not applicable.

Competing interests

The author declares no competing interests.

Funding

The author received no specific funding for this work.

Author Contribution

S.L. conceived the study, designed the architecture, implemented the experiments, analysed the results, and wrote the manuscript.

Acknowledgements

The author thanks the maintainers of the causal-learn, dagma, castle, and pgmpy projects for open-source implementations used in the baseline comparisons.

Data Availability

The Sachs, ALARM, and CHILD datasets are standard publicly available benchmark networks. The synthetic and multi-environment data, ground-truth adjacency matrices, trained-network configurations, held-out validation records, and experimental scripts used in this study are available from the corresponding author on reasonable request.

Code availability

The implementation of the bipolar architecture, network-grounded extraction and qualification, held-out conjunction validation, non-empty stability, FDR adjustment, baseline interfaces, and the scripts that generate every reported number are available from the corresponding author on reasonable request.

References

1. GDPR. *Regulation (EU) 2016/679* (General Data Protection Regulation). *Official J. Eur. Union* (2016).
2. Executive Office of the President. Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence. (2023).
3. Spirtes, P., Glymour, C. & Scheines, R. *Causation, Prediction, and Search* (MIT Press, 2000).
4. Chickering, D. M. Optimal structure identification with greedy search. *J. Mach. Learn. Res.* **3**, 507–554 (2002).
5. Zheng, X. et al. DAGs with NO TEARS: continuous optimization for structure learning. *Advances Neural Inform. Process. Systems* **31** (2018).
6. Bello, K., Aragam, B. & Ravikumar, P. DAGMA: learning DAGs via M-matrices and a new log-determinant acyclicity characterization. *Advances Neural Inform. Process. Systems* **35** (2022).
7. Mackie, J. L. Causes and conditions. *Am. Philos. Q.* **2**, 245–264 (1965).

8. Dul, J. Necessary condition analysis (NCA): logic and methodology of necessary but not sufficient causality. *Organizational Res. Methods*. **19**, 10–52 (2016).
9. Ragin, C. C. *Redesigning Social Inquiry: Fuzzy Sets and Beyond* (University of Chicago Press, 2008).
10. Pearl, J. *Causality: Models, Reasoning, and Inference* (Cambridge University Press, 2009).
11. Shimizu, S. et al. A linear non-Gaussian acyclic model for causal discovery. *J. Mach. Learn. Res.* **7**, 2003–2030 (2006).
12. Glymour, C., Zhang, K. & Spirtes, P. Review of causal discovery methods based on graphical models. *Front. Genet.* **10**, 524 (2019).
13. Li, S. Auditable machine learning for high-stakes decisions. *Res. Sq.* preprint at (2026). https://doi.org/10.21203/rs.3.rs-10966878/v1
14. Sachs, K. et al. Causal protein-signaling networks derived from multiparameter single-cell data. *Science* **308**, 523–529 (2005).
15. Mooij, J. M. et al. Distinguishing cause from effect using observational data: methods and benchmarks. *J. Mach. Learn. Res.* **17**, 1–102 (2016).
16. Heckerman, D., Geiger, D. & Chickering, D. M. Learning Bayesian networks: the combination of knowledge and statistical data. *Mach. Learn.* **20**, 197–243 (1995).
17. Benjamini, Y. & Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Royal Stat. Soc. B*. **57**, 289–300 (1995).
18. Peters, J., Bühlmann, P. & Meinshausen, N. Causal inference by using invariant prediction. *J. Royal Stat. Soc. B*. **78**, 947–1012 (2016).
19. Beinlich, I. A., Suermondt, H. J., Chavez, R. M. & Cooper, G. F. The ALARM monitoring system: a case study with two probabilistic inference techniques for belief networks. *Proc. 2nd Eur. Conf. Artificial Intelligence in Medicine* 247–256 (1989).
20. Cooper, G. F. et al. An evaluation of machine-learning methods for predicting pneumonia mortality. *Artif. Intell. Med.* **9**, 107–138 (1997).
21. Spiegelhalter, D. J., Dawid, A. P., Lauritzen, S. L. & Cowell, R. G. Bayesian analysis in expert systems. *Stat. Sci.* **8**, 219–247 (1993).

Figures

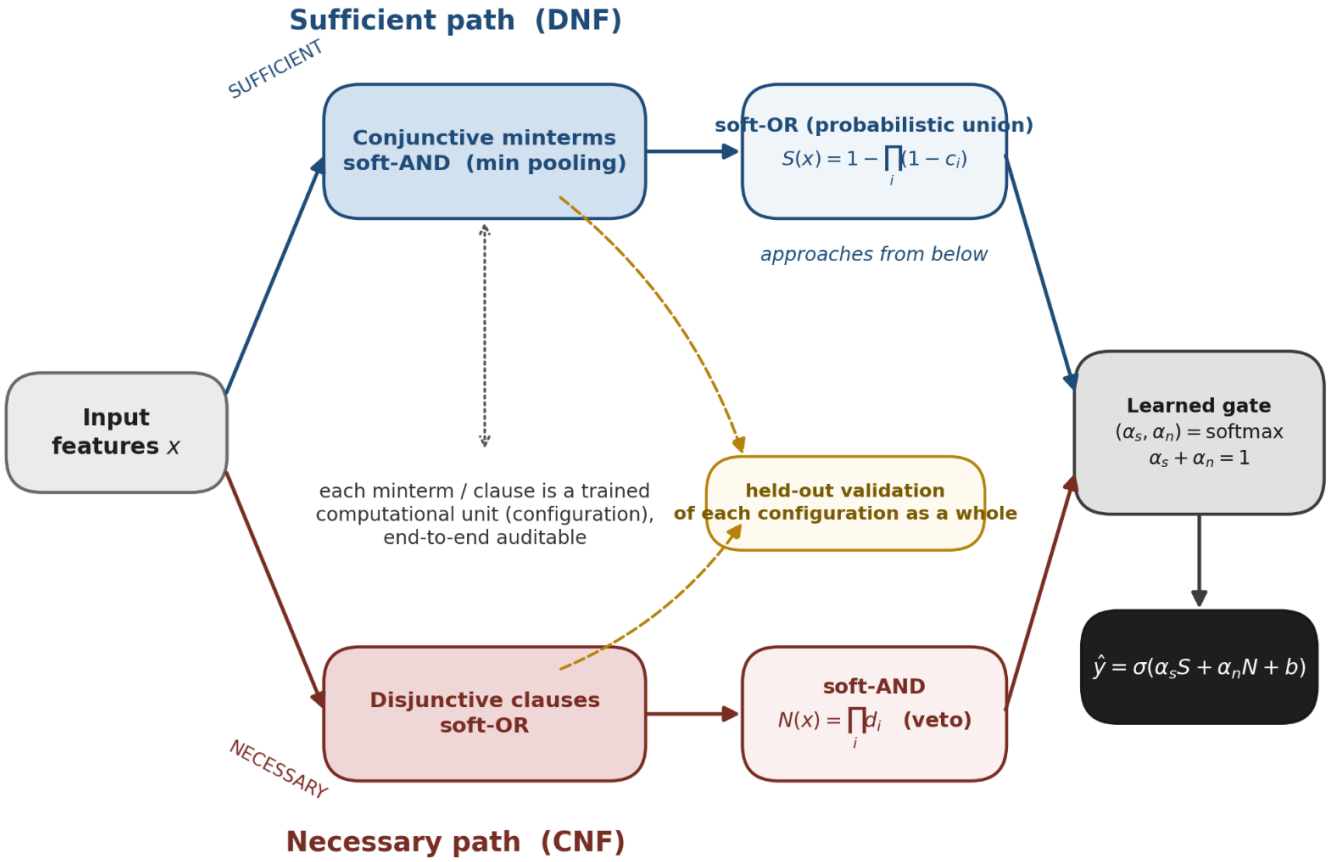


Figure 1

Bipolar Logic Network architecture. Input features feed two interpretable logical paths—a sufficient path in disjunctive normal form and a necessary path in conjunctive normal form—fused by a learned gate; every reported configuration is a trained minterm or clause, not a post-hoc label, and each configuration is validated as a whole on held-out data.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupplementaryInformation10.2.docx