ABLE: Choosing Perturbation Experiments to Recover Gene Logic

Yin Jun Phua [1] **Foo Wei Ten** [2]

Abstract

Scientific knowledge requires claims stated formally, checked against evidence, and paired with what remains undecided. Perturb-seq and CRISPR screens promise genome-scale interventional data, yet current machine-learning tools return ranked edge lists rather than executable regulatory logic. We address this for Boolean gene regulation under an idealized *in silico* Boolean intervention oracle, a setting where three capabilities are each necessary and none suffices alone. A neural proposer amortizes candidate-rule search at biological scale, replacing combinatorial enumeration with a sub-second forward pass. A symbolic verifier, LIFT-CERT, issues support-conditional uniqueness certificates on the declared regulator support and abstains explicitly when the data do not determine a rule. A coverage-guided active loop then closes the remaining gaps by naming the exact missing input combination. ABLE (Active Boolean Learning Engine) realizes these capabilities as a single propose–verify–query loop. Trained once on synthetic data, ABLE delivers support-conditional certified recovery across published biological Boolean models and certifies every rule in curated biological networks from modest warm starts. ABLE thus instantiates neuro-symbolic methodology for AI4Science, showing that the verifier and active loop, not the neural architectural prior, earn certification and yield a recoverability map where every rule is a checkable claim or a named missing observation.

1. Can Interventional Data Identify a Boolean Model?

Science is a verification discipline. A high-scoring prediction is not yet scientific knowledge until it is stated in the domain's own language and paired with justification that others can independently check. Neural networks have transformed scientific prediction, but their outputs are opaque and cannot certify correctness or specify what data are missing. Neuro-symbolic architectures address this gap by preserving formal, domain-grounded structure throughout the inference pipeline. The neural component supplies scale and pattern recognition, while the symbolic component ensures that every output is a checkable claim. We realize this principle for Boolean gene regulation.

Gene regulation offers a sharp instance of this challenge. Cells make discrete fate decisions (divide, differentiate, or die). Gene regulatory networks govern these decisions, with transcription factors, kinases, and signaling molecules interacting to produce specific outputs (Barabási & Oltvai, 2004). Predicting the effect of a drug or a genetic perturbation requires more than knowing which genes interact; it requires knowing exactly how regulators combine to produce an output. This is the model-building problem of turning observations into an executable, predictive model of a biological system.

Boolean networks are an established abstraction for this task. Each gene is modeled as on or off, updated at each time step by a logical rule over its regulators (Kauffman, 1969). Despite this coarse discretization, Boolean models correctly predict cell-fate attractors in yeast (Li et al., 2004), segment polarity in *Drosophila* (Albert & Othmer, 2003), drug synergies in cancer (Flobak et al., 2015), and signal transduction logic in mammalian cells (Saez-Rodriguez et al., 2009). The dominant structural class in published models is nested canalizing functions (NCFs), hierarchical priority rules where one dominant input can override all others (Jarrah et al., 2007; He & Macauley, 2016). This matches biological reality. A single transcription factor can silence a gene regardless of other signals. A recent meta-analysis of published Boolean network models found that 94.4% of investigated update rules are NCFs (Kadelka et al., 2024). NCFs are expressive enough to capture published regulatory logic, yet discrete enough to ask whether interventional data determine a rule exactly. This is a rare setting where mechanistic hypotheses are both biologically grounded and formally checkable.

Current ML tools for gene regulation produce ranked in-

[1]School of Computing, Institute of Science Tokyo, Tokyo, Japan [2]Berlin Institute of Health at Charité – Universitätsmedizin Berlin, Berlin, Germany. Correspondence to: Yin Jun Phua <phua@comp.isct.ac.jp>, Foo Wei Ten <foo-wei.ten@bih-charite.de>.

Accepted by ICML 2026 AI for Science Workshop.

teraction lists (network topology), not the executable logic that determines how a perturbation propagates (Aibar et al., 2017; Huynh-Thu et al., 2010). Recovering this logic exactly would allow perturbation screens (Dixit et al., 2016) to yield verified executable models rather than edge rankings. The real scientific question is not whether a rule fits the data, but whether interventional data identify it uniquely. Answering this requires both scalable search and exact elimination of alternatives. This is not a trade-off to manage but a neuro-symbolic problem to solve. Learned search is what traverses the hypothesis space at biological scale, while symbolic verification certifies when the data leave exactly one consistent rule and specifies what evidence is still missing. Classical symbolic enumeration via Learning From Interpretation Transitions (LFIT) (Inoue et al., 2014), GULA (Ribeiro et al., 2022) and its polynomial-time variant PRIDE (Ribeiro et al., 2021), guarantees completeness but its cost grows combinatorially with network size, exhausting reasonable time budgets by n=12; pure neural methods scale but cannot prove correctness. The combination is not optional; it is what the scientific question demands.

As opaque models grow more capable at prediction, the question of whether those predictions constitute scientific knowledge becomes more urgent, not less. Boolean gene regulation makes this concrete. The outputs are explicit NCF rules, the verifier can certify uniqueness, and unresolved cases return the exact missing evidence for targeted follow-up. Scope: we study an idealized *in silico* Boolean intervention oracle (set the full gene state, observe the synchronous next state), which is stronger than current Perturb-seq or CRISPR protocols; ABLE asks a foundational identification question under this oracle, with the gap to wet-lab deployment discussed in Section 7.

ABLE instantiates this template as a closed propose–verify–query loop whose outputs (Sections 3–6) form a recoverability map. For the AI4Science community, this work delivers:

1. A neural proposer that makes candidate-rule search tractable at biological scale, converting an otherwise combinatorial enumeration into a sub-second forward pass in a regime where classical symbolic LFIT enumeration is already infeasible (Appendix D.2, Appendix F).

2. A symbolic verifier, LIFT-CERT, that turns each proposed rule into a support-conditional uniqueness certificate when the observations leave no alternative Boolean function on the declared regulators, and abstains explicitly otherwise.

3. A coverage-guided active loop that, whenever LIFT-CERT abstains, names the exact regulator-input combination whose observation would close the remaining ambiguity; the three components together deliver support-conditional certified recovery across 31 published biological Boolean models and a per-gene recoverability map.

2. How Does the Propose–Verify–Query Pipeline Work?

ABLE converts perturbation data into a verified Boolean model through a propose–verify–query loop.

Boolean networks and NCFs. We observe $m \ll 2^n$ state transitions from a synchronous Boolean network and target exact recovery of every gene's update rule. Each state transition corresponds to one perturbation experiment. We set the system to a specific state and observe the next state.

A nested canalizing function (NCF), an ordered priority rule where the first matching condition determines the output, governs each gene in published Boolean models. If transcription factor A is active, the gene turns on regardless of other signals; otherwise check B; and so on. The formal definition appears in Appendix A.

A concrete example. Consider a gene G with two regulators, A and B. The update rule fires in order. If A=1, then G turns on; otherwise, if B=0, then G turns off; otherwise G turns on. This is an NCF, a short priority rule that checks regulators in order. Because G depends on only two regulators, there are 2^2=4 possible regulator-input combinations. Observing G's output for all four combinations fully determines the rule. No other Boolean function on $\{A, B\}$ can reproduce the same outputs. If even one combination remains unseen, an alternative rule could still differ on that unobserved row.

A	B	G	Reason
1	0	1	A=1 (first rule fires)
1	1	1	A=1 (first rule fires)
0	0	0	B=0 (second rule fires)
0	1	1	default

Each gene depends on k_i regulators, creating 2^{k_i} possible regulator-input combinations. Observing all 2^{k_i} combinations uniquely determines the function. LIFT-CERT *certifies* a rule when no alternative Boolean function on the same regulators remains consistent with the observations. The proposer is NCF-biased, but the verifier's uniqueness test is over all Boolean functions on the declared support, so certification applies to non-NCF rules as well.

Because each gene depends on only k_i out of n variables, identification is local. Observation cost scales with $\sum_i 2^{k_i}$ (the sum of local truth-table sizes) rather than 2^n. We write $k_{\max} = \max_i k_i$ for the largest regulator count in the network. The TLR5-Signaling pathway illustrates this locality. With n=40 genes and $k_{\max}$=3, most genes have only one

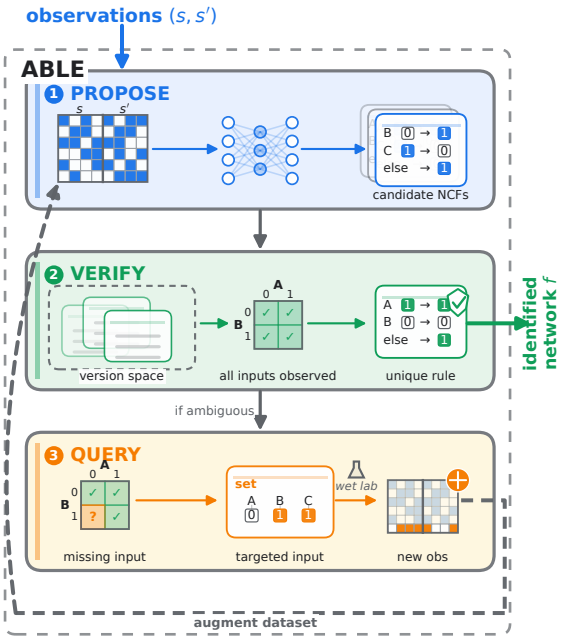

Figure 1. The ABLE propose–verify–query loop. The diagram shows candidate rules proposed from observed transitions, LIFT-CERT checking uniqueness, and unresolved ambiguity triggering targeted perturbation requests that feed new observations back into the loop.

or two regulators, giving $\sum_i 2^{k_i} \approx 150$. Roughly 260 observations therefore suffice for a certificate, rather than $2^{40} \approx 10^{12}$.

Pipeline overview. The pipeline has three steps:

1. **Propose candidate rules.** A set-transformer encoder (Lee et al., 2019), following the transition-encoding architecture of Phua and Inoue (Phua & Inoue, 2024), paired with an NCF-structured pointer decoder (Vinyals et al., 2015) processes observed transitions and proposes H=8 candidate update rules per gene, including regulator identity and canalizing structure.

2. **Verify and certify.** LIFT-CERT determines whether the observations narrow the set of consistent rules to exactly one. If so, it issues a uniqueness certificate for that rule (Appendix B).

3. **Request more data (optional).** If ambiguity remains, the system identifies which regulator-input combinations are still unobserved and requests targeted queries to discriminate among competing hypotheses.

When targeted follow-up is unavailable, the D4 cascade (a four-phase neural-guided shortlist search; full spec in Appendix E) performs restricted search over shortlist supports and nearby swap neighborhoods to resolve additional targets. Unlike LIFT-CERT, D4 issues only heuristic recovery, not a uniqueness certificate.

Training and benchmarks. We train the model once on synthetic NCF networks (n=50, $k_{\max}$=6) and apply it without retraining to all biological evaluations. For Table 2, we use a smaller n=15 checkpoint to ensure a fair comparison with symbolic baselines (GULA, PRIDE) that cannot scale to n=50; all other results use the n=50 model. We evaluate on 31 models from the Biodivine Boolean Models (BBM) repository (Pastva et al., 2023) ($n \leq 50$, $k_{\max} \leq 6$), four curated biological networks (3–12 genes), and a 17-model subset for observation-mode comparisons. We track two primary metrics and two additional diagnostics:

- *Per-gene exact recovery*: fraction of genes recovered correctly.
- *Whole-model exact recovery*: fraction of models with all genes correct.
- *Best-in-class match*: fraction of global states where the composition of per-gene closest NCF approximations correctly predicts the next state.
- *Per-gene coverage*: fraction of genes with all regulator-input combinations observed.

A target gene counts as per-gene exact when the predicted function on its declared regulator support, extended trivially over all variables, matches the ground-truth Boolean function on every state. We refer to this indicator as *certified-on-declared-support*, and use the stricter label *certified-and-exact* when the declared support also equals the true support. Headline tables in this paper report certified-on-declared-support, and Table 6 reports the per-k support-correctness audit on which the two indicators split. The gap is concentrated in $k \geq 4$ predictions where wrong-regulator errors begin to appear. Detailed metric and training specifications appear in Appendix A.

3. Does It Work on Real Published Biological Networks?

On the 31 BBM benchmark at m=500 with 10 repeats, 98.19% of issued certificates match the ground-truth rule exactly and the remaining 1.81% certify on a declared regulator support that differs from ground truth.

We evaluated ABLE across all 31 published biological Boolean models in BBM (Figure 2); for a detailed head-to-head comparison with symbolic baselines, we focus on four

Table 1. Scope of a LIFT-CERT uniqueness certificate: a statement about the data and the declared regulator support R, not about wet-lab realism.

LIFT-CERT certifies	LIFT-CERT does not certify
a unique Boolean rule on declared support R	that R is the true biological regulator set
all $2^{\lvert R\rvert}$ input rows are observed	that hidden regulators omitted from R have no effect
correctness under an idealized Boolean intervention oracle	that Perturb-seq or CRISPR realize that oracle
the missing input row when it abstains	that noisy, asynchronous, or continuous readouts are handled

curated biological networks: RAF kinase cascade (three genes), Wnt5a cell migration (seven genes), mammalian circadian rhythm (ten genes), and a 12-gene regulatory network ("Gene regulation", abbreviated Gene reg. in tables), which includes non-NCF rules and is chosen to stress the proposer's NCF inductive bias.

Table 2 compares the neural proposer alone (δLFIT2) against two symbolic baselines, GULA and PRIDE, and then under ABLE's active loop, at three passive observation budgets $m \in \{50, 200, 500\}$. High per-gene accuracy can mask poor whole-model accuracy. On Circadian at m=200, δLFIT2 reaches 0.93 per-gene but only 0.33 whole-model exact recovery. Gene reg. (n=12, $k_{\max}$=4) nevertheless recovers faster than Circadian (n=10, $k_{\max}$=6), confirming that maximum indegree, not network size, governs passive recovery difficulty (Section 5). ABLE's active loop certifies every rule on all four networks at every warm-start budget, achieving pg=1.00 and wm=1.00 across all 36 runs. The pg and wm columns measure prediction correctness against ground truth, while ABLE's matching cells additionally carry a LIFT-CERT uniqueness certificate, so the +q column reports the cost of certification rather than the cost of correctness. A δLFIT2 cell at 1.00/1.00 records that the proposer's predictions happen to match ground truth, while the matching ABLE cell with $+q > 0$ records the additional observations required to prove those rules are the only ones consistent with the data, within the declared regulator support and the NCF hypothesis class. The +q column is zero for RAF at any budget, up to $\sim$210 for Gene reg. at m=50, and decreases monotonically as the warm start grows. Even from the smallest m_0=50 warm start, ABLE reaches certification on every network using fewer total observations than the m=500 passive baselines need to even partially match whole-model recovery.

Ablating the NCF prior and the neural proposer. Because NCFs dominate published Boolean rules (Kadelka et al., 2024), a natural concern is that the NCF inductive bias, not the rest of the pipeline, drives recovery. Two matched ablations on the same 31 BBM models at m=500 (Appendix F) disentangle two design choices that are often conflated. Ablation B swaps the NCF pointer decoder for an unconstrained 2^k truth-table head while keeping the neural proposer fixed; it isolates the NCF inductive bias in the decoder head and finds that this bias does not gate certification (99.40% vs. 99.43%, below run-to-run noise), so the NCF prior is an ablatable design choice at biological scale. Ablation A swaps the neural proposer for NCF-aware combinatorial enumeration while keeping LIFT-CERT and the active loop identical; the combinatorial path exceeds the neural path on whole-model exact recovery on a 17-model paired cohort where exhaustive enumeration still fits a compute budget (99.4% vs. 80.6%, paired Δ = +18.8 percentage points (pp), 95% CI [+12.9, +25.3]; per-variable search-budget asymmetry noted in Appendix F, Table 12), but that regime does not extend to biological scale, where classical symbolic enumeration is already infeasible (Appendix D.2). The neural proposer, the symbolic LIFT-CERT verifier, and the coverage-guided active loop therefore form a three-way complementarity in which each component does something the other two cannot, and support-conditional certified recovery at biological scale requires all three.

Passive-only rescue when queries are unavailable. When targeted queries are impractical, the D4 cascade provides a query-free fallback for targets that neither the neural proposer nor LIFT-CERT has already resolved (Appendix E). At m=50 (see Table 8 for per-result checkpoints), post-D4 per-gene exact recovery rises from 40% to 93% on RAF and from 75% to 98% on Gene regulation. D4 reaches 100% whole-model exact recovery without queries at m=100 for three networks and m=200 for Circadian, consistent with the higher $k_{\max}$=6 difficulty analyzed in Section 5. D4's accept-as-correct flag has low precision in the hardest low-budget regime. On Circadian at m=50, only 17.1% of D4's proposed rules match ground truth (full results in Appendix E).

Across all 31 published models from the Biodivine repository, ABLE achieves 77.6% per-gene exact recovery from just 50 observations, rising to the 95.7%/51.0% headline figures at m=500 (Figure 2; Table 2). The improvement is sharpest for models with $k_{\max} \leq 4$; Section 5 analyzes where and why it degrades.

4. What Kinds of Measurements Are Actually Informative?

We report *Target PVE* (target-variable per-variable exact-recovery rate: the fraction of target genes whose rule is recovered exactly) and *Canonical PVE* (the fraction of tar-

Table 2. Capability ladder on four curated biological networks at three passive observation budgets $m \in \{50, 200, 500\}$ (3 seeds per cell; n=15 checkpoint, see Section 2). Columns pg and wm report per-gene exact recovery and whole-model exact recovery respectively; a gene counts as recovered only if a unique prediction matches ground truth on all 2^k regulator-input combinations. The +q column (ABLE only) reports the mean extra observations the active loop requested beyond the m-observation warm start to certify every rule; ± denotes standard deviation across seeds, omitted when zero. Passive baselines have no targeted follow-up phase (empty +q cells). GULA entries for Gene reg. show – (timeout >300 s). The light rule within each budget group separates ABLE from the passive baselines. Cells report *certified-on-declared-support*, where a target gene counts as exact when the predicted function on its declared regulator support matches the ground-truth Boolean function on every state. The strict *certified-and-exact* indicator additionally requires the declared support to equal the true support. Table 6 reports the per-k support-correctness audit on which the two split, and Section 2 states the joint definition.

		RAF (n=3)			Wnt5a (n=7)			Circadian (n=10)			Gene reg. (n=12)		
Budget	Method	pg	wm	+q	pg	wm	+q	pg	wm	+q	pg	wm	+q
m=50	GULA	1.00	1.00		.00	.00		.00	.00			–	
	PRIDE	1.00	1.00		.33	.00		.20	.00		.17	.00	
	δLFIT2	1.00	1.00		.90	.33		.50	.00		.94	.33	
	ABLE	1.00	1.00	0	1.00	1.00	149±24	1.00	1.00	194±4	1.00	1.00	210±3
m=200	GULA	1.00	1.00		.00	.00		.00	.00			–	
	PRIDE	1.00	1.00		.81	.33		.27	.00		.33	.00	
	δLFIT2	1.00	1.00		1.00	1.00		.93	.33		1.00	1.00	
	ABLE	1.00	1.00	0	1.00	1.00	64	1.00	1.00	118±13	1.00	1.00	131±3
m=500	GULA	1.00	1.00		.00	.00		.00	.00			–	
	PRIDE	1.00	1.00		1.00	1.00		.27	.00		.39	.00	
	δLFIT2	1.00	1.00		1.00	1.00		1.00	1.00		1.00	1.00	
	ABLE	1.00	1.00	0	1.00	1.00	0	1.00	1.00	48±16	1.00	1.00	55±10

get genes whose per-variable oracle achieves zero error, an information-theoretic ceiling on any identification algorithm given the data). Oracle accuracy measures next-state prediction accuracy on held-out random states. Infeasibility rate is the fraction of targeted queries whose requested state is unreachable under the chosen observation mode, and is zero by construction under diverse perturbations.

On a 17-model BBM subset (Figure 3), diverse perturbations achieve 99.2% per-gene exact recovery versus 36.2% for time-series trajectories and 51.0% for local-neighborhood perturbations (full comparison in Appendix Table 4).

Why time series fail through attractor confinement. Boolean networks converge to a small number of stable states (attractor basins), so trajectory observations revisit the same regions rather than covering new regulator-input combinations. A reachability census on all 31 models at m=500 quantifies this: trajectory observations cover all regulator-input combinations for only 85.1% of genes and 12.9% of models (Table 7).

This is not a simulation artifact. A 50-variable network with a handful of attractors concentrates observations in a vanishing fraction of 2^{50} possible states, and adding more trajectories yields diminishing returns. Under trajectory observations, even the targeted follow-up mechanism degrades, and 65.4% of requested perturbation states are unreachable from the trajectory-accessible state space. Any method that needs full coverage of regulator-input combinations therefore faces a fundamental data-efficiency barrier under native dynamics.

Local perturbations are a proposer limitation, not a data limitation. Local-neighborhood perturbations (changing one or two variables at a time from observed states) achieve 100% per-gene coverage and 100% certifiability under Hamming-2 (Table 7), so the information content is sufficient for uniqueness certification. Yet end-to-end recovery reaches only ∼51% per-gene exact recovery, because we trained ABLE on synthetic data, and the proposer performs poorly on this correlated data geometry. Adapting the proposer to this geometry is a clear engineering opportunity, not a fundamental barrier, in contrast to trajectory and Hamming-1 modes where coverage itself is incomplete.

5. Where Is the Practical Complexity Boundary?

Given diverse perturbation data, the observation cost of certified recovery empirically tracks total local truth-table rows $\sum_i 2^{k_i}$ rather than network size n (the locality scaling law, an empirical fit; Figure 4, Appendix B). In practice, the hardest gene dominates this sum, so $k_{\max}$, the number of regulators of the most-connected gene, is the single quantity a practitioner needs to predict recovery.

The practical lookup rule. A regulator audit across 31 models (Appendix Table 5) reveals a sharp complexity

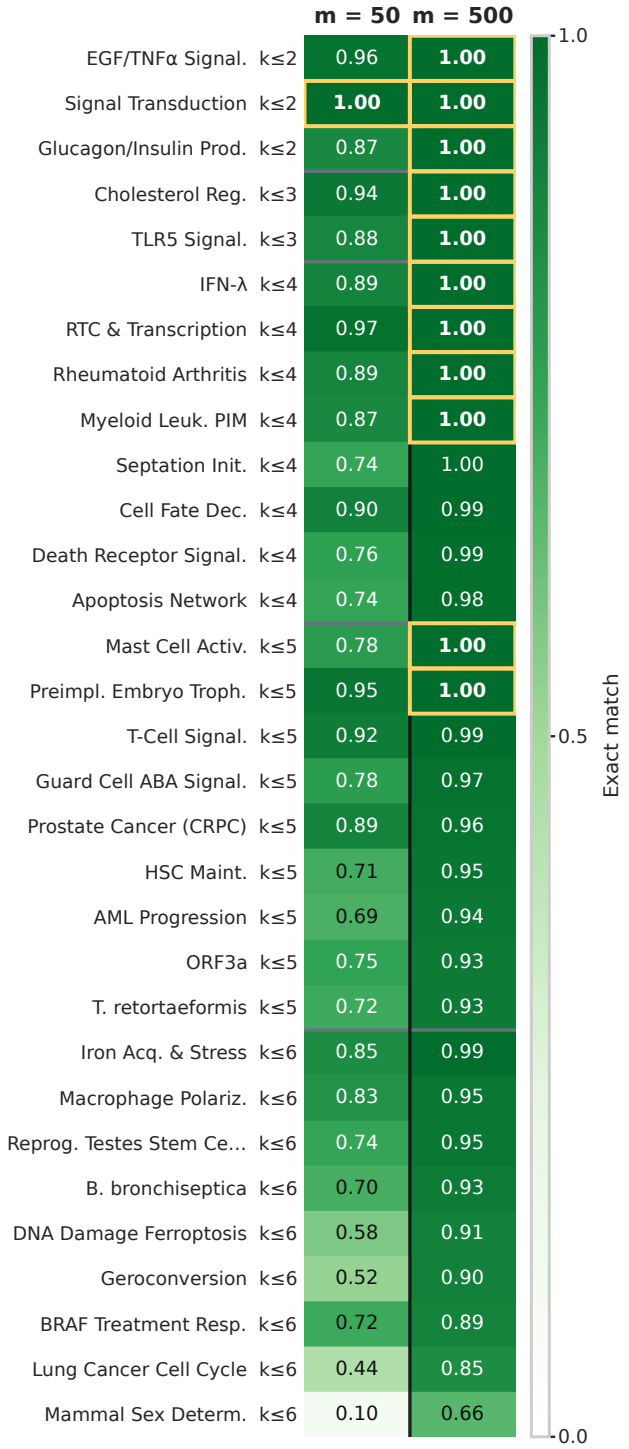

Figure 2. Per-gene exact recovery across 31 published biological Boolean models at two observation budgets (m=50 vs. m=500). Rows are ordered by $k_{\max}$; horizontal lines separate groups; gold outlines mark 100% recovery.

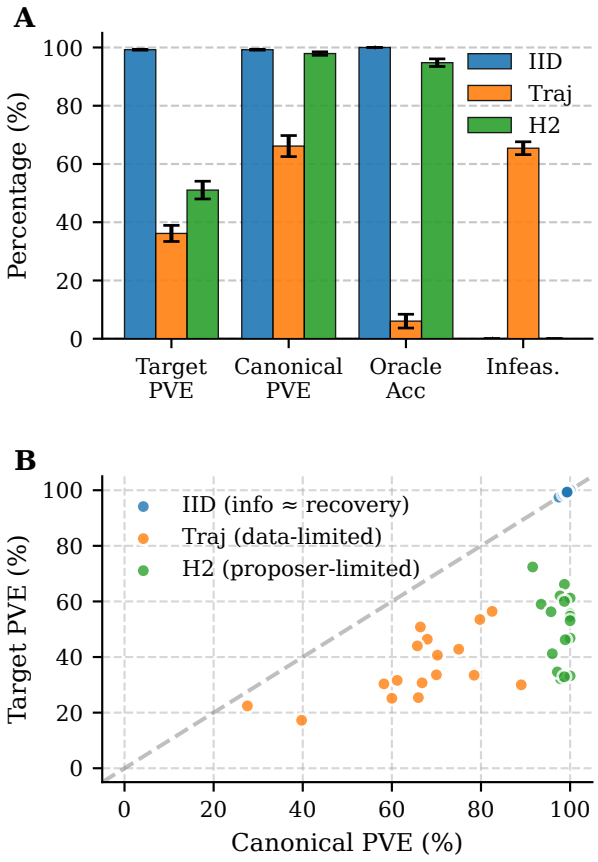

Figure 3. Per-gene exact recovery on 17 BBM models under diverse perturbations (IID), time-series trajectories (Traj), and local-neighborhood perturbations at Hamming distance 2 (H2). **(A)** Mean Target PVE, Canonical PVE, oracle accuracy, and infeasibility rate by observation mode. **(B)** Per-model scatter in (Canonical PVE, Target PVE) space; the dashed $y{=}x$ line marks the per-variable oracle ceiling, separating data-limited regimes (low Canonical PVE) from proposer-limited regimes (gap below the diagonal).

boundary. Genes with $k \leq 3$ achieve perfect recovery across all 8,120 predictions, with zero wrong-regulator risk. At $k = 4$ recovery stays reliable (>90% exact recovery) with occasional wrong-regulator errors, at $k = 5$ roughly half of predictions omit a true regulator, and $k \geq 6$ sits outside the method's reliable regime. The bottleneck is regulator-support identification. Selecting k true inputs from over 40 candidates becomes combinatorially harder as k increases.

Whole-model recovery depends on the hardest gene. Because per-gene errors compound across all variables, whole-model exact recovery hinges on the hardest gene in the network. Grouping the 31 models by $k_{\max}$ confirms this at the whole-model level. All models with $k_{\max} \leq 3$ achieve 100% whole-model exact recovery, while recovery drops to ≈9% at $k_{\max} = 6$ as wrong-regulator errors

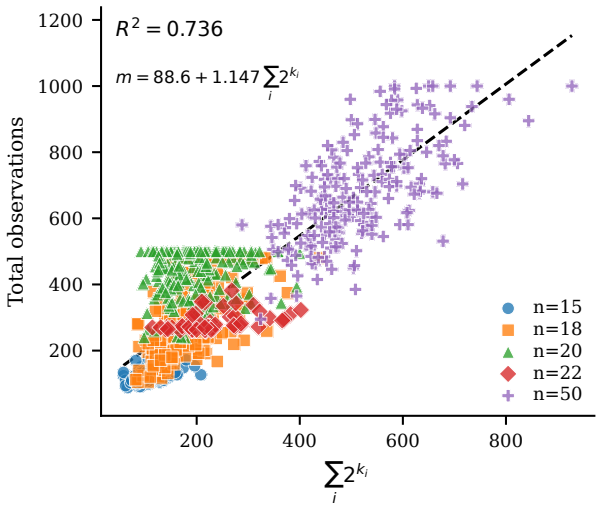

Figure 4. Empirical locality scaling trend across five network scales (n=15 to n=50). Observations to certified recovery vs. total local truth-table rows ($\sum_i 2^{k_i}$). Linear fit: $m_{\text{cert}} = 88.6 + 1.147 \sum_i 2^{k_i}$ ($R^2 = 0.736$). Local rule complexity ($\sum_i 2^{k_i}$), not network size n, empirically drives observation cost.

accumulate (Appendix Table 5 gives the full breakdown).

Network size barely matters. The empirical locality scaling trend (Figure 4) predicts that recoverability depends on local rule complexity rather than network size, and the BBM benchmark confirms this (Figure 2). A large low-k network (TLR5-SIGNALING, n=40, $k_{\max}$=3) achieves perfect recovery, while a smaller high-k network (GEROCONVERSION, n=23, $k_{\max}$=6) drops substantially; the largest network (AML-PROGRESSION, n=45, $k_{\max}$=5) still recovers most rules (exact values in Figure 2). In practice, practitioners can estimate $k_{\max}$ from regulatory topology databases such as KEGG (Kanehisa & Goto, 2000) or Reactome (Milacic et al., 2024) before running the pipeline. The Kadelka et al. meta-analysis (Kadelka et al., 2024) places most published biological networks within the $k_{\max} \leq 4$ favorable regime.

The observed fraction of state space drops from 3.8×10^{-3} at n=15 to 6.1×10^{-13} at n=50, yet support-conditional certified recovery succeeds at every scale (Figure 4; Appendix B).

How many observations do I need? Recovery improves steadily with observation budget but with diminishing returns. Across 31 BBM models, whole-model exact recovery rises from 3.2% (m=50) to 51.0% (m=500), with saturation at lower budgets for $k_{\max} \leq 4$ (Appendix Table 5).

Table 3 summarizes the current performance envelope across observation modes and rule complexities.

Table 3. Capability map: current recovery performance by observation mode and local rule complexity ($k_{\max}$). **Strong**: >90% whole-model exact recovery. **Partial**: >90% per-gene exact recovery, 30–90% whole-model. **Weak**: >90% per-gene exact recovery, <30% whole-model. **Poor**: <90% per-gene exact recovery.

	Diverse perturb.	Local-nbhd.	Trajectory
$k_{\max} \leq 4$	Strong	Partial	Poor
$k_{\max} = 5$	Partial	Poor	Poor
$k_{\max} \geq 6$	Weak	Poor	Poor

6. When Do Targeted Follow-Up Experiments Help?

When an initial random screen leaves some rules unresolved, targeted follow-up can close the gap, with the size of the gain depending on rule complexity.

To see why targeted queries matter, consider two candidate four-input rules that agree on 15 of 16 regulator-input combinations but differ on exactly one unobserved combination. Random perturbations mostly revisit the 15 states the two candidates already share, so ambiguity persists indefinitely. One targeted query to the missing combination resolves the disagreement immediately. In paired-rule experiments of this kind on n=50 networks, the neural proposer without targeting resolves only 46.5% of such cases (consistent with a coin flip), while one targeted query achieves 100% resolution (Appendix D). *Which* state the pipeline queries matters more than *how many* it queries.

Start broad, target selectively. On the full 31-model BBM benchmark at m=500 (Appendix Figure 5), targeted follow-up improves per-gene exact recovery from 95.7% to 98.3% ($\Delta = +2.64$ pp; $p = 0.047$, Wilcoxon signed-rank, one-sided $p = 0.024$). The improvement concentrates in complex networks (Appendix Figure 6). For $k_{\max} = 6$, targeted follow-up improves whole-model exact recovery from 8.9% to 63.3% and per-gene exact recovery from 89.1% to 98.4%. For simple networks ($k_{\max} \leq 4$), random perturbation experiments already saturate recovery, so reallocating part of the fixed budget to targeted queries can shift the data distribution and reduce performance; for $k_{\max} = 2$–3, targeted follow-up reduces whole-model exact recovery from 100% to 93.3% and 80.0% respectively.

The practical design rule is to start with a broad random screen and use targeted follow-up only for networks where the initial screen reveals unresolved genes with high regulator counts (≥5). To validate this rule, we treat each model's $k_{\max}$ as a gating signal, routing models with $k_{\max} \leq \tau$ to the passive m=500 screen and the rest to targeted follow-up; because each model is evaluated independently under either policy in the underlying experiments, we score any threshold τ by mix-and-matching the existing per-model

passive and active runs without re-training. Sweeping τ on a 17-model training cohort selects τ=4, raising per-gene exact recovery from 99.23% to 99.41%, and freezing τ=4 on a disjoint 14-model held-out cohort lifts per-gene exact recovery from 92.85% to 93.97% (post-hoc holdout-optimal τ=5 reaches 95.44%; Appendix Figure 7).

Why targeted queries must be paced into re-fitting rounds. On synthetic n=50 systems ($k_{\max}$=4, budget B=1500), an uncapped targeted follow-up policy achieves 100% recovery but certifies only 5% of rules; pacing into B_r=75 batches restores 93% certified at lower total cost (1,260 vs. 1,495 observations; Appendix Table 9). The learner must update its hypothesis between batches; without re-fitting, queries target stale regulator guesses and waste budget on already-covered combinations. The full analysis, including paired-rule constructions that isolate this effect, appears in Appendix D.

7. How Does This Relate to Existing ML Methods?

Gene regulatory network inference. Unlike ranked edge-list inference (GENIE3 (Huynh-Thu et al., 2010), GRNBoost2 (Moerman et al., 2019), DeepSEM (Shu et al., 2021)), ABLE returns the exact executable rule per gene or specifies the additional observation that would resolve the remaining ambiguity, given Boolean interventional data.

Perturbation biology and the data landscape. ABLE's observation budgets fit modern Perturb-seq screens (Replogle et al., 2022); its intervention assumption (full-state Boolean clamp) is stronger than deployed CRISPR protocols (Table 1).

Why Boolean models? Unlike continuous ODE parameterizations, which suffer from fundamental identifiability problems even with extensive data (Gutenkunst et al., 2007), Boolean rules are short, independently verifiable, and cover the dominant regulatory logic class in published models (Section 1).

Exact Boolean-network and NCF identification. Classical algebraic enumeration of consistent NCF or Boolean models from a fixed dataset (Hinkelmann et al. (Hinkelmann et al., 2011), Dimitrova et al. (Dimitrova et al., 2011)), constraint-based ensemble inference from transcriptomic data (Chevalier et al., 2025), and Monte Carlo tree search over Boolean models from phenotype constraints (Glazer et al., 2023) all become infeasible at biological scale (Appendix D.2). ABLE adds targeted query selection, amortized neural search, and formal uniqueness certificates, making recovery tractable in the regime covering most published biological networks.

Neuro-symbolic propose–verify systems. Unlike propose–verify systems for geometry proofs (AlphaGeometry (Trinh et al., 2024)) or program synthesis (DreamCoder (Ellis et al., 2021)), and unlike neural–symbolic recovery of continuous gene regulatory network dynamics via symbolic regression (Yu et al. (Yu et al., 2026)), ABLE targets exact Boolean rule recovery with per-rule uniqueness certificates, building on the set-transformer (Lee et al., 2019) and transition-encoding architecture of Phua and Inoue (Phua & Inoue, 2024) with an NCF pointer decoder and a verifier that drives a certificate-based scientific loop.

Causal inference and active experiment design. Unlike adaptive experiment design for causal discovery (Eberhardt et al., 2005; Sándor & Antal, 2025), which orients edges in a causal graph via do-operations that break confounding (Pearl, 2009), ABLE's coverage-guided planner targets truth-table completion for each gene, connecting to classical exact learning of Boolean functions with few relevant inputs (Angluin, 1988; Bshouty & Costa, 2018) extended to a multi-gene, batched setting.

8. Discussion

Certificates as scientific safeguards. ABLE's uniqueness certificates provide an auditable record of which mechanistic claims the data determine and which remain ambiguous. When the system cannot issue a certificate, it specifies the missing evidence and requests a targeted query rather than guessing. This prove-or-abstain discipline gives a governance mechanism for AI-assisted scientific reasoning, since every claimed rule is either formally justified or explicitly flagged as provisional.

Limitations and future directions. All evaluations are *in silico*, and bridging the intervention-type gap (Section 7, Table 1) is the main step toward wet-lab deployment. Three directions extend the current results. Prospective validation on real Perturb-seq data (Dixit et al., 2016) already fits within ABLE's observation budgets. A continuous-to-Boolean interface would let the pipeline consume RNA-seq or protein-level readouts directly. Pushing reliable recovery beyond $k_{\max}$=4 requires architectural refinements or hybrid symbolic–neural search over regulator supports.

The neuro-symbolic case for science. ABLE instantiates the neuro-symbolic paradigm as a three-way rather than two-way partnership, where each of the three components does something the other two cannot. A LIFT-CERT uniqueness certificate states that the data determine a rule exactly within the declared regulator support, a stronger guarantee than a confidence score attached to a model.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Numbers 25K21269 and 25K03190. This study was carried out using the TSUBAME4.0 supercomputer at Institute of Science Tokyo.

References

Aibar, S., González-Blas, C. B., Moerman, T., Huynh-Thu, V. A., Imrichova, H., Hulselmans, G., Rambow, F., Marine, J.-C., Geurts, P., Aerts, J., van den Oord, J., Atak, Z. K., Wouters, J., and Aerts, S. SCENIC: single-cell regulatory network inference and clustering. *Nature Methods*, 14(11):1083–1086, 2017. doi: 10.1038/nmeth.4463.

Albert, R. and Othmer, H. G. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in Drosophila melanogaster. *Journal of Theoretical Biology*, 223(1):1–18, 2003. doi: 10.1016/s0022-5193(03)00035-3.

Angluin, D. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988. doi: 10.1007/BF00116828.

Barabási, A.-L. and Oltvai, Z. N. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, 2004. doi: 10.1038/nrg1272.

Bshouty, N. H. and Costa, A. Exact learning of juntas from membership queries. *Theoretical Computer Science*, 742: 82–97, 2018. doi: 10.1016/j.tcs.2017.12.032.

Chevalier, S., Becker, J., Gui, Y., Noël, V., Su, C., Jung, S., Calzone, L., Zinovyev, A., del Sol, A., Pang, J., Sinkkonen, L., Sauter, T., and Paulevé, L. Data-driven inference of Boolean networks from transcriptomes to predict cellular differentiation and reprogramming. *npj Systems Biology and Applications*, 11:105, 2025. doi: 10.1038/s41540-025-00569-z.

Dimitrova, E. S., García-Puente, L. D., Hinkelmann, F., Jarrah, A. S., Laubenbacher, R., Stigler, B., Stillman, M., and Vera-Licona, P. Parameter estimation for Boolean models of biological networks. *Theoretical Computer Science*, 412(26):2816–2826, 2011.

Dixit, A., Parnas, O., Li, B., Chen, J., Fulco, C. P., Jerby-Arnon, L., Marjanovic, N. D., Dionne, D., Burks, T., Raychowdhury, R., Adamson, B., Norman, T. M., Lander, E. S., Weissman, J. S., Friedman, N., and Regev, A. Perturb-Seq: Dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens. *Cell*, 167(7):1853–1866.e17, 2016. doi: 10.1016/j.cell.2016.11.038.

Eberhardt, F., Glymour, C., and Scheines, R. On the number of experiments sufficient and in the worst case necessary to identify all causal relations among N variables. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 178–184. AUAI Press, 2005.

Ellis, K., Wong, C., Nye, M. I., Sablé-Meyer, M., Morales, L., Hewitt, L. B., Cary, L., Solar-Lezama, A., and Tenenbaum, J. B. DreamCoder: bootstrapping inductive program synthesis with wake-sleep library learning. In *PLDI '21: 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, pp. 835–850. ACM, 2021. doi: 10.1145/3453483.3454080.

Flobak, Å., Baudot, A., Remy, E., Thommesen, L., Thieffry, D., Kuiper, M., and Lægreid, A. Discovery of drug synergies in gastric cancer cells predicted by logical modeling. *PLOS Computational Biology*, 11(8):e1004426, 2015. doi: 10.1371/journal.pcbi.1004426.

Glazer, B. J., Lifferth, J. T., and Lopez, C. F. Automatic mechanistic inference from large families of Boolean models generated by Monte Carlo tree search. *Frontiers in Cell and Developmental Biology*, 11:1198359, 2023. doi: 10.3389/fcell.2023.1198359.

Gutenkunst, R. N., Waterfall, J. J., Casey, F. P., Brown, K. S., Myers, C. R., and Sethna, J. P. Universally sloppy parameter sensitivities in systems biology models. *PLoS Computational Biology*, 3(10):e189, 2007. doi: 10.1371/journal.pcbi.0030189.

He, Q. and Macauley, M. Stratification and enumeration of Boolean functions by canalizing depth. *Physica D: Nonlinear Phenomena*, 314:1–8, 2016. doi: 10.1016/j.physd.2015.09.016.

Hinkelmann, F., Brandon, M., Guang, B., McNeill, R., Blekherman, G., Veliz-Cuba, A., and Laubenbacher, R. C. ADAM: analysis of discrete models of biological systems using computer algebra. *BMC Bioinformatics*, 12:295, 2011. doi: 10.1186/1471-2105-12-295.

Huynh-Thu, V. A., Irrthum, A., Wehenkel, L., and Geurts, P. Inferring regulatory networks from expression data using tree-based methods. *PLoS ONE*, 5(9):e12776, 2010. doi: 10.1371/journal.pone.0012776.

Inoue, K., Ribeiro, T., and Sakama, C. Learning from interpretation transition. *Machine Learning*, 94(1):51–79, 2014. doi: 10.1007/s10994-013-5353-8.

Jarrah, A. S., Raposa, B., and Laubenbacher, R. Nested canalyzing, unate cascade, and polynomial functions. *Physica D: Nonlinear Phenomena*, 233(2):167–174, 2007. doi: 10.1016/j.physd.2007.06.022.

Kadelka, C., Butrie, T.-M., Hilton, E., Kinseth, J., Schmidt, A., and Serdarevic, H. A meta-analysis of Boolean network models reveals design principles of gene regulatory networks. *Science Advances*, 10(2):eadj0822, 2024. doi: 10.1126/sciadv.adj0822.

Kanehisa, M. and Goto, S. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1): 27–30, 2000. doi: 10.1093/nar/28.1.27.

Kauffman, S. A. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3):437–467, 1969. doi: 10.1016/0022-5193(69)90015-0.

Lee, J., Lee, Y., Kim, J., Kosiorek, A. R., Choi, S., and Teh, Y. W. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pp. 3744–3753, 2019.

Li, F., Long, T., Lu, Y., Ouyang, Q., and Tang, C. The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences*, 101(14):4781–4786, 2004. doi: 10.1073/pnas.0305937101.

Milacic, M. et al. The Reactome pathway knowledgebase 2024. *Nucleic Acids Research*, 52(D1):D672–D678, 2024. doi: 10.1093/nar/gkad1025.

Moerman, T., Aibar Santos, S., Bravo González-Blas, C., Simm, J., Moreau, Y., Aerts, J., and Aerts, S. GRNBoost2 and Arboreto: efficient and scalable inference of gene regulatory networks. *Bioinformatics*, 35(12):2159–2161, 2019. doi: 10.1093/bioinformatics/bty916.

Pastva, S., Šafránek, D., Beneš, N., Brim, L., and Henzinger, T. Repository of logically consistent real-world Boolean network models. *bioRxiv*, 2023. doi: 10.1101/2023.06.12.544361. Preprint.

Pearl, J. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, UK, 2nd edition, 2009. ISBN 978-0521895606.

Phua, Y. J. and Inoue, K. Variable assignment invariant neural networks for learning logic programs. In *International Conference on Neural-Symbolic Learning and Reasoning*, pp. 47–61. Springer, 2024.

Replogle, J. M., Saunders, R. A., Pogson, A. N., Hussmann, J. A., Lenail, A., Guna, A., Mascibroda, L., Wagner, E. J., Adelman, K., Lithwick-Yanai, G., Iremadze, N., Oberstrass, F., Lipson, D., Bonnar, J. L., Jost, M., Norman, T. M., and Weissman, J. S. Mapping information-rich genotype-phenotype landscapes with genome-scale Perturb-seq. *Cell*, 185(14):2559–2575.e28, 2022. doi: 10.1016/j.cell.2022.05.013.

Ribeiro, T., Folschette, M., Magnin, M., and Inoue, K. Polynomial algorithm for learning from interpretation transition. In *Proceedings of the 1st International Joint Conference on Learning & Reasoning (IJCLR)*, Virtual, Greece, 2021. URL https://hal.science/hal-03347026.

Ribeiro, T., Folschette, M., Magnin, M., and Inoue, K. Learning any memory-less discrete semantics for dynamical systems represented by logic programs. *Machine Learning*, 111(10):3593–3670, 2022. doi: 10.1007/s10994-021-06105-4.

Saez-Rodriguez, J., Alexopoulos, L. G., Epperlein, J., Samaga, R., Lauffenburger, D. A., Klamt, S., and Sorger, P. K. Discrete logic modelling as a means to link protein signalling networks with functional analysis of mammalian signal transduction. *Molecular Systems Biology*, 5:331, 2009. doi: 10.1038/msb.2009.87.

Sándor, D. and Antal, P. Efficient structure learning of gene regulatory networks with Bayesian active learning. *BMC Bioinformatics*, 26:150, 2025. doi: 10.1186/s12859-025-06149-6.

Shu, H., Zhou, J., Lian, Q., Li, H., Zhao, D., Zeng, J., and Ma, J. Modeling gene regulatory networks using neural network architectures. *Nature Computational Science*, 1(7):491–501, 2021. doi: 10.1038/s43588-021-00099-8.

Trinh, T. H., Wu, Y., Le, Q. V., He, H., and Luong, T. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024. doi: 10.1038/s41586-023-06747-5.

Vinyals, O., Fortunato, M., and Jaitly, N. Pointer networks. In *Advances in Neural Information Processing Systems 28 (NeurIPS)*, pp. 2692–2700, 2015.

Yu, Z., Ding, J., and Li, Y. Discovering network dynamics with neural symbolic regression. *Nature Computational Science*, 6(2):156–168, 2026. doi: 10.1038/s43588-025-00893-8.

A. Formal Setup and Notation

This appendix collects formal definitions and notation used throughout the paper.

Boolean networks. A synchronous Boolean network with n genes has state $x(t) \in \{0,1\}^n$ and update rules $f_i : \{0,1\}^n \to \{0,1\}$ such that $x_i(t{+}1) = f_i(x(t))$. We observe $m \ll 2^n$ state transitions (s_t, s_{t+1}) and target exact functional recovery: recover $\hat{f}_i = f_i$ for all $x \in \{0,1\}^n$.

In practice, each state transition corresponds to one perturbation experiment. We set the system to a state (e.g., by forcing specific variables on or off) and observe the next state.

Nested canalizing rules. An NCF on regulators $(r_1, \ldots, r_k)$ is parameterized by canalizing inputs $a \in \{0,1\}^k$ and outputs $b \in \{0,1\}^{k+1}$:

$$f(x) = \begin{cases} b_1, & x_{r_1} = a_1, \\ b_2, & x_{r_1} \neq a_1,\ x_{r_2} = a_2, \\ \vdots & \\ b_{k+1}, & \text{otherwise.} \end{cases} \tag{1}$$

Detailed metric definitions. **Per-gene exact recovery**: a gene counts as recovered only when the predicted rule equals the ground-truth rule on all 2^{k_i} regulator-input combinations. Partial matches on a strict subset of the truth table do not contribute. **Whole-model exact recovery**: for each network this indicator is 1 when every gene is exactly recovered and 0 otherwise; the reported figure is the mean over the evaluation set of networks. **Best-in-class match**: fraction of global states where the composition of per-gene closest NCF approximations correctly predicts the next state. **Per-gene coverage**: fraction of genes with all regulator-input combinations observed.

Training details. Diverse perturbation observations use idealized full-state interventions that are substantially stronger than current CRISPR or Perturb-seq protocols, which typically perturb sparse subsets of genes. Trajectory observations correspond to time series from native dynamics. Local-neighborhood perturbations (Hamming distance 2) correspond to single or double interventions from observed states.

B. Certification Theory and Algorithm

Recovery means the pipeline outputs a rule that matches ground truth. A LIFT-CERT uniqueness certificate is stronger but narrower: it proves that, within the declared regulator support, the observations leave no alternative. A certified rule is the *only* Boolean function on its declared support consistent with the data, so correctness within the declared support follows from the data alone, without access to ground truth. Certification does not by itself rule out a function of additional regulators outside the declared support; Definition 1 below makes this support-conditional scope precise. This distinction matters because a method can recover the right answer without being able to certify it (Appendix D shows this gap concretely: 100% recovery but only 5% certified).

Formal definition. Definition 1 (LIFT-CERT uniqueness certificate). A LIFT-CERT uniqueness certificate for variable i with declared support R states that the version space $\mathcal{V}(R)$ (the set of all Boolean functions on R consistent with the observations) is a singleton. The certificate is support-conditional: it guarantees uniqueness within R, but not global correctness if the true support extends beyond R.

Why is the cost exactly 2^k? Proposition 1 (Exact Boolean uniqueness-certificate cost). Fix variable i and declared support R with $|R| = k$. The unrestricted Boolean version space $\mathcal{V}(R)$ is a singleton if and only if every regulator-input bin $z \in \{0,1\}^k$ has been observed at least once. Equivalently:

$$\inf_{\mathcal{A}} \sup_{g:\{0,1\}^k \to \{0,1\}} \min\{m : |\mathcal{V}_m(R)| = 1\} = 2^k. \tag{2}$$

Sketch. One observation from each bin reveals all 2^k truth-table entries. If any bin is unobserved, two rules that differ only on that bin remain indistinguishable.

Because $\mathcal{V}^{\mathrm{NCF}}(R) \subseteq \mathcal{V}(R)$, a singleton unrestricted version space immediately implies at most one consistent NCF. This is why LIFT-CERT uses the unrestricted Boolean version space rather than the NCF-restricted one. The unrestricted version yields a stronger uniqueness certificate, not a weaker one, at the same 2^k cost.

How does this scale system-wide? The local-complexity view yields an empirical whole-system observation-cost trend:

$$m_{\mathrm{cert}} = 88.6 + 1.147 \sum_i 2^{k_i}, \qquad R^2 = 0.736. \tag{3}$$

The intercept reflects the initial random seed budget; the slope slightly above one is coupon-collector overhead. We refer to this relation as an empirical locality scaling trend rather than a scaling law; the R^2 summarizes a linear fit on five network scales, not a proven bound. The observation cost for a uniqueness certificate empirically scales with summed local truth-table size, not ambient dimension 2^n.

This empirical trend also explains why the BBM biological benchmarks (Section 3) are tractable: the regression predicts roughly 260 observations for the TLR5-Signaling pathway, consistent with the 224–290 observed in practice.

Algorithm 1 Paced uniqueness-certificate pipeline

Require: Oracle $\mathcal{O}$; seed budget m_0; round cap B_r; max rounds T
Draw m_0 initial transitions $\rightarrow$ dataset $\mathcal{D}$
for $r = 1$ **to** T **do**
  for each uncertified variable i **do**
    Propose candidate supports/rules from $\mathcal{D}$; keep best $(\hat{R}_i, \hat{f}_i)$
    Compute uncovered bins U_i under $\hat{R}_i$
  end for
  Build up to B_r targeted queries from pooled uncovered bins *// pacing cap*
  Query oracle; augment $\mathcal{D}$
  for each uncertified i with singleton $\mathcal{V}_{\mathcal{D}}(\hat{R}_i)$ stable for 2 rounds **do**
    Run swap-one adversarial audit; if passed, issue LIFTCERT uniqueness certificate
  end for
  if all certified **then**
    break
  end if
end for
Run omitted-regulator audit on certified set *// empirical safeguard, not a formal proof of global correctness*
return $\{\hat{f}_i\}$ with per-variable support-conditional uniqueness certificates

Table 4. Three observation modes on 17 BBM models (mean ± SE). Per-gene exact = per-gene rule recovery. Whole-model exact = all genes correct. Best-in-class = whole-model state accuracy under closest per-gene NCF approximation. Queries unreachable = follow-up queries targeting unreachable states. Cells report *certified-on-declared-support*, where a target gene counts as exact when the predicted function on its declared regulator support matches the ground-truth Boolean function on every state. The strict *certified-and-exact* indicator additionally requires the declared support to equal the true support. Table 6 reports the per-k support-correctness audit on which the two split, and Section 2 states the joint definition.

Mode	Per-gene exact	Best-in-class	Whole-model exact	Queries unreachable
Diverse	**99.2%**	**100.0%**	**80.6%**	**0.0%**
Trajectory	36.2%	6.0%	0.0%	65.4%
Local-nbhd.	51.0%	94.8%	0.6%	0.0%

What loop implements this? Algorithm 1 summarizes the full pipeline.

C. Extended Benchmark Details

This appendix collects detailed tables and figures supporting the results in the main text.

Observation-Mode Comparison. Table 4 compares three observation modes on 17 BBM models, supporting Section 4.

Budget Sweep. Table 5 reports aggregate performance across observation budgets on 31 BBM models.

Table 5. Budget sweep on 31 published biological Boolean models (passive mode; m=500 averaged over 10 repeats per model, lower budgets single-seed). All numbers are aggregate means. Per-gene exact = per-gene exact recovery. Whole-model exact = whole-model exact recovery (all genes correct). Best-in-class = whole-model state accuracy under closest per-gene NCF approximation. Cells report *certified-on-declared-support*, where a target gene counts as exact when the predicted function on its declared regulator support matches the ground-truth Boolean function on every state. The strict *certified-and-exact* indicator additionally requires the declared support to equal the true support. Table 6 reports the per-k support-correctness audit on which the two split, and Section 2 states the joint definition.

Budget m	Per-gene exact	Whole-model exact	Best-in-class
50	77.6%	3.2%	23.1%
100	86.3%	19.4%	48.1%
200	92.0%	32.3%	77.5%
500	95.7%	51.0%	94.2%

Table 6. Regulator-set prediction quality by number of regulators (k) on 31 BBM models (10 repeats, m=500). Exact = predicted regulators match ground truth. Harmful = omitted regulator causes output errors. Output exact = fraction of predictions with correct function output on all inputs.

Regulators (k)	# pred.	% exact	% harmful	% out exact
1	4,380	99.5	0.0	100.0
2	2,570	99.6	0.0	100.0
3	1,170	97.4	0.0	100.0
4	660	87.9	7.6	92.4
5	430	46.7	48.6	50.9
6	140	18.6	81.4	17.9
Total	9,350	94.8	4.0	96.0

Support-correctness audit underlying the headline split. Table 6 reports the per-k support-correctness audit on the same 31 BBM models at m=500, separating two indicators that the headline columns combine. *% exact* is the fraction of predictions whose declared regulator set equals the true support, and *% out exact* is the fraction whose function on the declared support reproduces the ground-truth Boolean function on every input. The headline tables report *certified-on-declared-support*, which collapses to *% out exact* per row. The strict *certified-and-exact* indicator additionally conditions on *% exact*, so its per-k value is upper-bounded by *% exact*.

Full Per-Model Recovery Heatmap. Figure 5 shows the full per-model recovery heatmap across all observation budgets.

Observation-Mode Certifiability Diagnosis. Table 7 reports the certifiability analysis from a reachability census at m=500 across all 31 BBM models. These are information-theoretic limits independent of the algorithm: a gene is certifiable when the observations cover all 2^{k_i} regulator-input

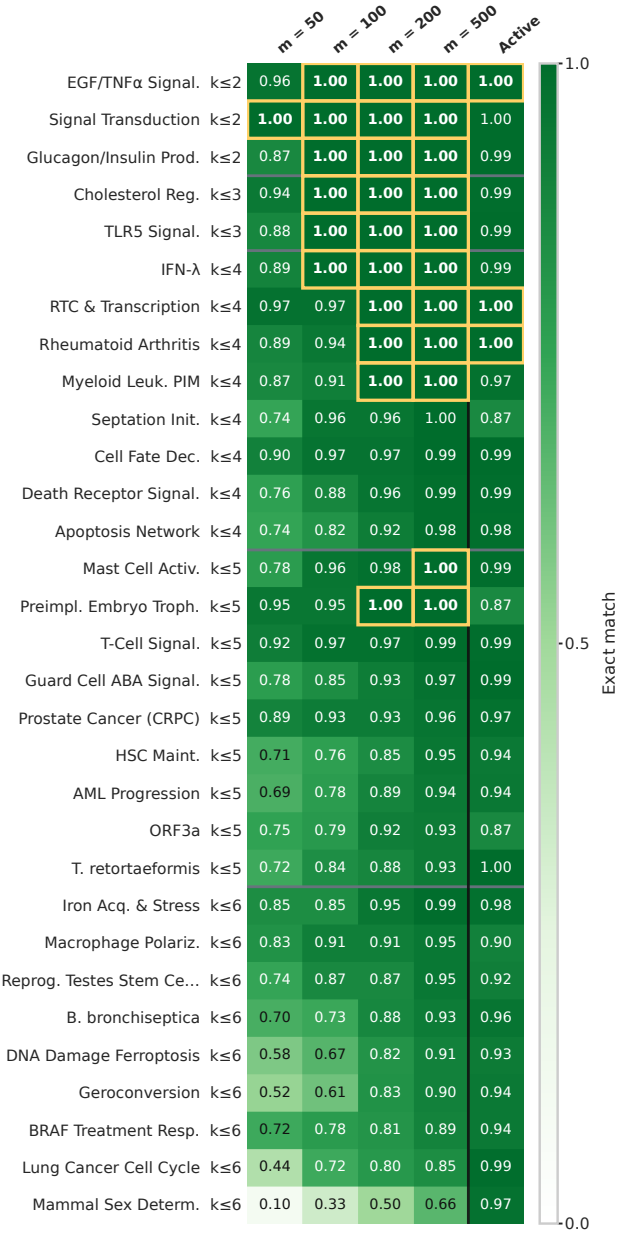

Figure 5. Full per-model recovery heatmap across 31 published biological Boolean models. Columns show passive recovery at budgets $m \in \{50, 100, 200, 500\}$ and targeted follow-up. Models are ordered by $k_{\max}$, then by descending m=500 recovery. Gold outlines mark 100% recovery.

combinations without contradiction.

Protocol audit. Table 8 summarizes, for each headline table or figure in the main text, which proposer checkpoint is used, which warm-start observation budgets are evaluated, and whether the run is passive (no targeted queries) or active (passive warm start followed by coverage-guided follow-up).

Table 7. Certifiability analysis at m=500 (10 replicates, 31 BBM models). Certifiable = all 2^{k_i} regulator-input combinations observed. These are information-theoretic limits independent of the algorithm.

Mode	Per-gene certified	Whole-model certified	Per-gene coverage	Diagnosis
Diverse perturbations	100.0%	100.0%	100.0%	Fully certifiable
Hamming-2	100.0%	100.0%	100.0%	Fully certifiable (algorithm-limited)
Hamming-1	97.6%	77.4%	99.8%	Severely data-limited
Trajectory	85.1%	12.9%	95.0%	Severely data-limited

Table 8. Protocol audit for headline tables and figures. n identifies the proposer checkpoint used (by network size); active means the paced LIFT-CERT loop of Algorithm 1 is invoked after the passive warm start.

Reference	n	Warm start m	Mode
Table 2	15	$\{50, 200, 500\}$	passive + active
Fig. 2	50	$\{50, 500\}$	passive
Fig. 5	50	$\{50, 100, 200, 500\}$	passive + active
Fig. 3	50	500	passive
Fig. 4	$\{15, 18, 20, 22, 50\}$	varies	active
Table 12	50	500	active

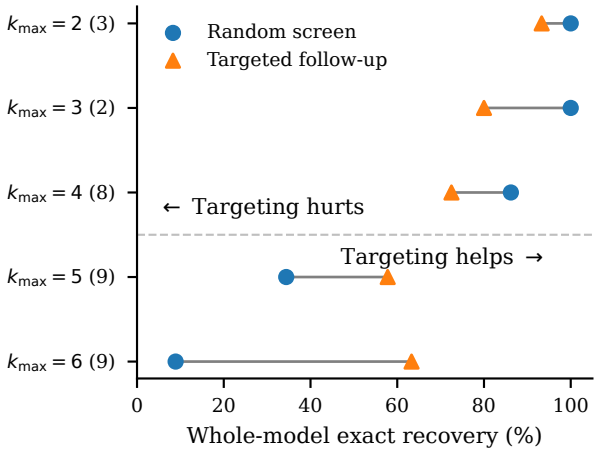

Figure 6. Whole-model exact recovery under a random screen and targeted follow-up, stratified by $k_{\max}$. The crossover occurs near $k_{\max} = 5$: targeted follow-up is lower for simpler networks and higher for more complex ones.

D. Adaptivity Analysis and Symbolic Baselines

D.1. Why Does Adaptivity Collapse Without Query Pacing?

The mechanism behind the modest pooled gain from targeted follow-up (Section 6) becomes clearer on controlled synthetic n=50 systems: targeted follow-up can recover the exact model yet fail to *certify* it if too much budget is spent before the learner updates its support estimate.

Can one missing bin really matter? We construct 200 adversarial k=4 twin pairs on n=50-variable networks, where each pair differs on exactly one of 16 support bins and produces identical labels on all other observations. A passive

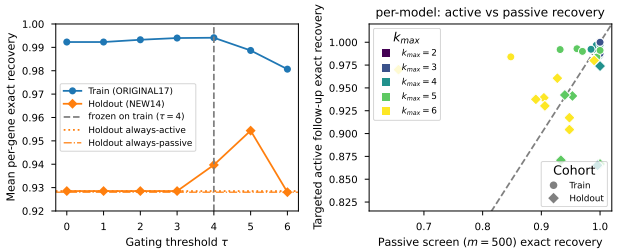

Figure 7. Sensitivity of the complexity-aware gating policy to the threshold τ. Per-gene exact recovery is shown for the 17-model training cohort and the disjoint 14-model held-out cohort as τ varies, alongside always-passive (m=500) and always-active baselines. The vertical marker at τ=4 indicates the threshold selected on the training cohort.

Table 9. Adaptivity collapse on synthetic n=50 trajectory-knockout systems ($k_{\max}$=4, budget B=1500): targeted follow-up restores uniqueness certificates and lowers observation cost.

Policy	Recovery	Certified	Rounds	Obs.
Uncapped ($B_r = \infty$)	100%	5%	2.9	1495
Paced ($B_r = 75$)	100%	93%	15.7	1260

neural proposer resolves only 46.5% of cases (consistent with a coin flip), while one targeted query to the hidden bin resolves all 200 (100%). One untargeted query to an already-observed bin resolves 0%. The value lies in querying the *right* state, not merely querying more.

Why do stale batches fail? Proposition 2 (Stale-support batches cannot prove correctness). If round t queries only previously unseen bins of $\hat{R}_t$, with at most one query per bin, then that round cannot produce two observations sharing the same $\hat{R}_t$-projection but differing in output. Therefore, if $\hat{R}_t \neq R^\star$, even complete coverage of all $2^{|\hat{R}_t|}$ bins of $\hat{R}_t$ does not prove the rule correct; at least one additional feedback round is required.

Sketch. A one-shot batch over new $\hat{R}_t$-bins yields no repeated $\hat{R}_t$-projection, so the data never expose a disagreement attributable to a missing regulator.

What does pacing buy in practice? Table 9 compares an uncapped active-query policy with paced batches on synthetic n=50 systems. The main algorithmic lesson is that adaptivity concerns not only *which* states are queried, but also *when* the learner is forced to update its hypothesis.

D.2. Symbolic Baseline Comparison

Classical enumeration-based symbolic learners (GULA (Ribeiro et al., 2022) and PRIDE (Ribeiro et al., 2021)) either require complete observation of every state transition or over-generate vacuous rules from partial data, and their runtimes grow sharply with the number of variables. ABLE's amortized neural proposer instead returns candidate rules in sub-second time across the full range we evaluate, delegating correctness to the LIFT-CERT verifier. Table 10 reports the operating-regime contrast across observation regime, runtime scale, and exact-recovery behavior.

Table 10. Operating-regime comparison of GULA, PRIDE, and ABLE across observation regime, runtime scale, and exact-recovery behavior.

Method	Regime	Partial obs	n=10 runtime	n=40 runtime	n=50 runtime	Exact recovery
GULA	Complete obs	Over-generates (vacuous)	42.7s	Infeasible	Infeasible	100% (complete)
PRIDE	Partial obs	Conservative	Minutes	Infeasible	Infeasible	70–97% (partial)
ABLE	Partial obs	Amortized	<1s	<1s	<1s	95.7% (500 obs)

Table 11. D4 cascade results on curated biological networks (5 seeds per condition). Neural = neural proposer per-gene exact recovery. LIFT-CERT = after LIFT-CERT certification. Post-D4 = after D4 passive rescue. D4 precision = fraction of D4 attempts that matched the benchmark rule. wm = whole-model exact recovery. Cells report *certified-on-declared-support*, where a target gene counts as exact when the predicted function on its declared regulator support matches the ground-truth Boolean function on every state. The strict *certified-and-exact* indicator additionally requires the declared support to equal the true support. Table 6 reports the per-k support-correctness audit on which the two split, and Section 2 states the joint definition.

Network	m	Neural	LIFT-CERT	Post-D4	D4 prec.	wm
RAF (n=3)	50	0.400	0.400	0.933	88.9%	80%
	100	0.533	0.800	1.000	100%	100%
	200	1.000	1.000	1.000	—	100%
	500	1.000	1.000	1.000	—	100%
Wnt5a (n=7)	50	0.629	0.629	0.971	92.3%	80%
	100	0.886	0.914	1.000	100%	100%
	200	0.943	1.000	1.000	—	100%
	500	1.000	1.000	1.000	—	100%
Circadian (n=10)	50	0.300	0.300	0.420	17.1%	0%
	100	0.400	0.400	0.640	40.0%	0%
	200	0.460	0.720	1.000	100%	100%
	500	0.780	0.980	1.000	100%	100%
Gene reg. (n=12)	50	0.750	0.750	0.983	93.3%	80%
	100	0.850	0.900	1.000	100%	100%
	200	0.983	1.000	1.000	—	100%
	500	1.000	1.000	1.000	—	100%

E. D4 Passive Cascade on Biological Networks

Table 11 reports the full D4 cascade results across all tested budgets and seeds on the four curated biological networks.

Important caveat. D4's internal uniqueness flag is local to its searched shortlist neighborhood, not a global uniqueness certificate. For example, in the Circadian network (m=200, seed 42, gene 6), D4 found one zero-conflict candidate in its restricted search window, while LIFT-CERT identified 39 globally valid observationally equivalent rules. We therefore report D4 as heuristic benchmark recovery driven by neural inductive bias, not as certified identification. Post-D4 recovery is monotonically non-decreasing relative to LIFT-CERT in this evaluation because the eval-

uation harness invokes D4 only on targets that LIFT-CERT leaves unresolved; this gated invocation is a property of the evaluation harness, not an algorithmic no-harm guarantee.

F. Ablation: Necessity of the Neural Proposer and the NCF Prior

Since 94.4% of published Boolean rules are NCFs (Kadelka et al., 2024), a central question is whether ABLE's recovery numbers are driven by the NCF architectural prior in the decoder rather than by the propose–verify–query loop. A single ablation cannot resolve this because removing the neural proposer and removing the NCF bias probe different subsystems. We therefore run two complementary ablations that each hold the rest of the pipeline fixed.

Ablation A: combinatorial proposer. We replace the neural proposer with NCF-aware combinatorial enumeration. For each target gene, we first prune the candidate regulator pool to the top-12 variables by mutual information (MI) with the target (using the same encoder features that seed the neural shortlist), then enumerate regulator supports S drawn from that pool of size $k \in \{1, \ldots, 4\}$, giving roughly $\sum_{k=1}^{4} \binom{12}{k} \approx 793$ candidate supports per variable; for each support we select the best-fit NCF by symbolic search. This MI-pruned enumeration is the run reported in Table 12; we refer to the un-pruned $k \in \{1, \ldots, k_{\max}\}$ enumeration over all $\binom{n}{k}$ supports (with $k_{\max}$=6) as the *conceptual* cost of the combinatorial proposer, whose scaling we discuss below. Everything downstream of the proposer is identical to the main pipeline: the same LIFT-CERT verifier, the same active loop, the same D4 rescue cascade, the same identifiability audit, and the same encoder features used to seed the shortlist. The benchmark protocol is also identical: the 31 BBM models, 10 repeats per model, and a passive warm start of m=500 observations.

Ablation B: unconstrained decoder. We replace the NCF pointer decoder, which emits a $2k_{\max}$-long canalizing (a, b) representation biased toward NCFs, with an unconstrained truth-table head that emits the full 2^k output vector of a Boolean function over the predicted support. The encoder, the training distribution (n=50 NCF data), and the optimizer are held fixed; the head change requires retraining from scratch on the same n=50 NCF training distribution. The verifier, the active loop, and the 31 BBM models at m=500 with 10 repeats are identical to the main pipeline.

Neural proposer trades audit for amortized search. Table 12 gives the matched 31-model Main-vs-A-vs-B comparison at strategy=multi with matched seeds across all 31 models; on that cohort, Ablation A exceeds the main pipeline by +0.75 pp per-gene exact recovery and +17.4 pp whole-model exact recovery, with 25/6/0 and 23/8/0 paired wins/ties/losses respectively. The cleanest paired statistic, with matched seeds, matched active-loop strategy, and roughly matched per-variable search budget, is available only on a 17-model subset; on that cohort, combinatorial enumeration reaches 99.4% whole-model exact recovery versus 80.6% for the neural proposer (paired Δ = +18.8 pp, 95% CI [+12.9, +25.3], N=17, 14/3/0 wins/ties/losses). Per-gene exact recovery on the same 17-model cohort is essentially saturated for both systems (99.99% vs. 99.24%; Δ = +0.75 pp), so the differentiating quantity is whole-model recovery. A known caveat of both comparisons is a per-variable search-budget asymmetry: the MI-pruned combinatorial run of Ablation A evaluates ∼793 supports per variable (all subsets of the top-12 MI regulators up to $k \leq 4$), whereas the neural proposer evaluates `beam_width= 8` candidates per variable, so part of the gap reflects candidate-pool size rather than proposer class. What the *un-pruned* conceptual enumeration cannot avoid is cost: without MI pruning, for every unresolved target it must expand $\sum_{k=1}^{k_{\max}} \binom{n}{k}$ supports at $k_{\max}$=6 and fit each symbolically. On the 31-model BBM evaluation at $k_{\max}$=6, the combinatorial proposer takes on the order of 100× longer wall time than the neural proposer per model, and Appendix D.2 (Table 10) documents how the gap widens with n. The neural path is the one that stays tractable as n grows; it is not the one that maximizes recovery when exhaustive search is affordable.

Table 12. Ablation: necessity of the neural proposer (Ablation A) and the NCF prior (Ablation B). Bigger is better for recovery and certification; smaller is better for observations. All three systems share the same LIFT-CERT verifier, the same active loop, and the same 31 BBM models at m=500 with 10 repeats. All cells come from a matched 31-model Main-vs-A-vs-B comparison, except the certified per-gene fraction, which uses a dedicated certification pass on the same 31 models at m=500 with 10 repeats, counting abstentions on unidentifiable gates as non-certified (`audit_omitted=True`). Cells report *certified-on-declared-support*, where a target gene counts as exact when the predicted function on its declared regulator support matches the ground-truth Boolean function on every state. The strict *certified-and-exact* indicator additionally requires the declared support to equal the true support. Table 6 reports the per-k support-correctness audit on which the two split, and Section 2 states the joint definition.

	ABLE (main) neural+NCF	Ablation A combinatorial	Ablation B unconstrained
Per-gene exact recovery	98.3%	99.1%	98.4%
Whole-model exact recovery	68.1%	85.5%	67.7%
Certified per-gene fraction	99.43%	99.57%	99.40%
Mean total observations	456	448	477

NCF prior does not gate certification. Ablation B isolates the inductive bias in the decoder head. Raw rule recovery is nearly indistinguishable from the main pipeline (98.4% vs. 98.3% per-gene, 67.7% vs. 68.1% whole-model), because the combinatorial rescue fallback still extracts exact

rules within the allowed budget whenever the data suffice. A prior version of this paper hypothesized that the NCF prior earns its place on the certificate side: that an unconstrained 2^k head would commit to supports too large for LIFT-CERT to close observationally within the same budget, so the verifier should abstain more often. A dedicated certification pass on the same 31 BBM models at $m{=}500$ with 10 repeats refutes this. Counting abstentions on unidentifiable gates as non-certified (`audit_omitted=True`), the unconstrained head certifies 99.40% of target genes against 99.43% for the NCF head, a difference well below run-to-run noise. The NCF prior's distinct role is a compact $2k_{\max}$-parameter pointer head that the neural proposer can learn to emit; substituting a $2^{k_{\max}}$-parameter truth-table head trains equally well at $n{=}50$ without loss of accuracy or certification. ABLE as deployed pairs the neural proposer with the NCF pointer head because that combination is the cheapest in both compute and parameter count at fixed accuracy and certification, not because either is required to earn the certificate.

Metric semantics: certification vs. exact recovery. Table 12 reports a certified-per-gene fraction (99.43%) that slightly exceeds the per-gene exact-recovery rate (98.3%). This $+1.1$ pp gap is expected from Definition 1's support-conditional scope. Two effects pull in opposite directions: (i) certified-but-not-exact, where a rule is certified over a declared support R that differs from ground truth (for instance, when regulators omitted from R happen to stay constant across the observed transitions, LIFT-CERT cannot detect the contradiction and certifies the restricted rule); and (ii) exact-but-not-certified, where the proposer predicts the correct rule but the data do not cover all $2^{|R|}$ input bins, so LIFT-CERT abstains and `audit_omitted=True` counts the gene as non-certified. The positive sign of the net gap indicates that (i) slightly outnumbers (ii) on the 31 BBM models.