

# Proof-Carrying Optimality for Finite Identification under Bounded Adversarial Answer Errors

Vikram Lex

[vikram@karlexai.com](mailto:vikram@karlexai.com)

KarLex AI

---

## Research Article

**Keywords:** exact learning, active learning, adversarial errors, query complexity, certifying algorithms, decision trees

**Posted Date:** September 20th, 2026

**DOI:** <https://doi.org/10.21203/rs.3.rs-11052076/v1>

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

[Read Full License](#)

**Additional Declarations:** Competing interest reported. Vikram Lex is the sole founder of KarLex AI, a C corporation with no external investors or revenue related to this work, and declares no other relevant financial or non-financial interests.

---

# Proof-Carrying Optimality for Finite Identification under Bounded Adversarial Answer Errors

Vikram Lex<sup>1\*</sup>

<sup>1\*</sup>KarLex AI, Newark, Delaware, USA.

Corresponding author(s). E-mail(s): [vikram@karlexai.com](mailto:vikram@karlexai.com);

## Abstract

Exact learning from queries is a model of machine learning in which a learner actively identifies an unknown hypothesis. We study its finite, realizable form when queries come from a declared alphabet and an adversary may corrupt a bounded number of answers. For this setting we develop independently checkable certificates of exact worst-case query complexity. Portable certificates combine a noiseless strategy with a lower witness to prove an affine identity for every error budget, without expanding the corrupted-answer game. An isolation witness consists of classes on which each permitted probe agrees on all but at most one class. We prove its exact value for arbitrary residual error budgets and match it to a known robustification bound. Expanded strategy and adversary certificates handle individual budgets; separate distance and rational-dual certificates establish non-adaptive optima. Isolation witnesses increase compact coverage from 4 to 9 of 15 primary tables and from 25 to 69 of 101 sweep tables. Combined evidence certifies 302 of 303 sweep cells at budgets zero, one, and two. An additional 256-table evaluation increases compact coverage from 71 to 161; constructed certificates reach 256 classes. As a learning example, identifying a monotone conjunction of four variables with two corrupted answers requires exactly 14 adaptive queries versus 20 non-adaptive queries. The contribution is a certification method for finite query learning and its computational study; the singleton-search and robustification formulas are established prior results.

**Keywords:** exact learning, active learning, adversarial errors, query complexity, certifying algorithms, decision trees

# 1 Introduction

A learner must identify one of finitely many hypotheses using queries from a fixed alphabet. An adversary may corrupt a bounded number of answers, with full knowledge of the learner’s strategy. We study certificates that establish the minimum worst-case number of queries for this problem.

This is an exact-learning problem in the theoretical foundations of machine learning (ML) and artificial intelligence (AI): the unknown behaviour is a hypothesis, the inputs are membership or value queries, and the answers are labels or program outputs. Query selection is active because each observed answer can affect the next input. The desired guarantee is complete identification within a specified finite hypothesis class, with an optimal worst-case number of queries despite bounded corruption. Section 3 gives the precise correspondence and a reproducible concept-learning example.

Bounded-error identification has a long history, from the Rényi–Ulam liar game [1, 2] through coping with errors in binary search [3], thresholds for a liar [4], computing with noisy information [5], and fault-tolerant decision trees [6, 7], to exact learning from unreliable membership queries [8] and noisy generalized binary search [9]. Surveys by Pelc [10] and the monograph of Cicalese [11] survey this literature. With binary outputs and no answer errors, our objective specializes to the minimum-height decision-tree problem for a fixed test set, studied by Garey [12], Laber and Nogueira [13], and Bshouty and Makhoul [14].

For an explicit finite table, optimal depth is the value of a minimax game whose state space grows with the number of classes, alphabet size, and error budget. A certificate allows an independent verifier to check a computed value without repeating that search. This follows the approach of *certifying algorithms* [15] and proof-carrying computation [16, 17]. Related examples include unsatisfiability proofs checked by DRAT-TRIM [18] and rational certificates for integer-programming results [19, 20].

## 1.1 Contributions

We make four main contributions. First, we develop two independently checkable certificate families for finite restricted-query games. Portable certificates combine an index-based noiseless strategy with a small lower witness and can prove an affine law for every error budget. Expanded certificates combine a fixed-budget strategy with an adversary directed acyclic graph (DAG) and can settle any cell for which the search and proof construction finish. We prove the portable shallow rules complete for the singleton, depth-one, and blind-triple depth-two regimes. We then generalise the lower witness to any number of classes whose complete-alphabet partitions isolate at most one class per probe. An exact unequal-residual potential proves the lower bound, and a witness of size  $d + 1$  matches any checked depth- $d$  noiseless tree. The resulting certificate is verified without witness discovery or minimax synthesis; bounded discovery distinguishes exhaustive absence at the requested size from a resource limit.

Second, we give a hierarchy of checkable lower bounds: pairwise, volume and subset-volume, blind-triple and isolation, truthful-path multicover, and protected-pair bounds. We also derive an exact non-adaptive baseline from the minimum-distance criterion, with rational dual certificates. Third, we give a direct equality reduction from Set

Cover for our many-valued table model. This recovers the known NP-hardness of the zero-error minimum-height problem while showing that the optimum of the constructed table equals the minimum cover size. Fourth, we evaluate every method on a fixed primary matrix and a fixed evaluation sweep. Together these contain 333 declared evaluation cells, of which 332 receive two-sided proofs after combining expanded and portable evidence. Compact laws cover 9 of 15 primary and 69 of 101 sweep tables. An additional 256-table random evaluation gives 161 compact laws against 71 for the shallow rules, and constructed structured certificates reach 256 classes. These populations are reported separately because they share families and some instances. Failed expanded runs remain explicit even when another certificate proves the value.

The robustification inequality and its binary decreasing-residual construction appear in Maharjan and Watson [7, Lemma 2], whose Observation 3 also gives the tight AND-function example. The singleton-search value  $M(e + 1) - 1$  is established as well: Meng et al. [21, Theorem 3(1) and Lemma 10] give its one-error case, and Fraknói et al. [22, Section 2, following Theorem 4] state it for arbitrary error budgets. We extend these arguments to table-bound upper and lower certificates in the finite many-valued model. Verification is polynomial in the explicit table and supplied proof; the limits on discovery and proof size are discussed in Section 14.

## 2 Finite Restricted-Query Identification

### 2.1 Behaviour Tables and Semantic Classes

Let  $\mathcal{H} = (h_1, \dots, h_N)$  be a finite catalog of deterministic executable hypotheses, each mapping a hashable input to a hashable output, and let  $\mathcal{Q} = (q_1, \dots, q_Q)$  be a finite ordered *declared probe alphabet*. Evaluating the catalog on the alphabet gives the raw matrix  $h_i(q_j)$ . Two hypotheses are *behaviourally indistinguishable* on  $\mathcal{Q}$  when their complete rows agree, which is an equivalence relation; its classes  $\mathcal{C} = (c_1, \dots, c_M)$  are the *semantic classes*, computed by iterative refinement one probe at a time. Since every member of a class agrees on every declared probe, the induced map

$$A : \mathcal{Q} \times \mathcal{C} \rightarrow \mathcal{Y}, \quad A(q, c) = h(q) \text{ for any } h \in c,$$

is well defined. We call  $A$  the *behaviour table* and work with it exclusively. All statements in this paper are about  $A$ , i.e. about semantic classes and the complete declared alphabet; syntactic duplicates in  $\mathcal{H}$  are irrelevant by construction, and the compression ratio  $N/M$  is reported but never used in a bound.

Each probe  $q$  induces a set partition  $\pi_q$  of  $\mathcal{C}$  by output value. Two probes with the same partition are interchangeable for everything that follows, which is the basis of a search reduction (Corollary 24) and of the output-free digest (Section 4). We write  $\pi_q|_L$  for the restriction of  $\pi_q$  to a subset  $L \subseteq \mathcal{C}$ , and say  $q$  *refines*  $L$  when  $\pi_q|_L$  has at least two blocks, and *separates*  $c, c'$  when  $A(q, c) \neq A(q, c')$ . We require throughout that every pair of distinct classes is separated by some declared probe; otherwise the game is unsolvable and every procedure in this paper rejects the input.

## 2.2 The Corrupted-Answer Game

**Definition 1** (Identification game) Fix a behaviour table  $A$  and an integer budget  $e \geq 0$ . An adversary secretly fixes a target  $c^* \in \mathcal{C}$ . In round  $t = 1, 2, \dots$  a deterministic learner selects a probe  $q_t \in \mathcal{Q}$  as a function of the transcript  $(q_1, y_1, \dots, q_{t-1}, y_{t-1})$  and receives an answer  $y_t$ . A finite transcript is *valid* for  $c^*$  if

$$|\{t : y_t \neq A(q_t, c^*)\}| \leq e.$$

A strategy *succeeds* if along every valid transcript it halts having named  $c^*$ . Its *depth* is the worst-case number of rounds, and

$$\text{OPT}_A(e) = \min\{\text{depth of } T : T \text{ succeeds}\}.$$

Three features distinguish this from the classical liar game and matter throughout. First, the alphabet is *restricted*: the learner may not ask an arbitrary subset question, only one of the  $Q$  declared probes, and repetition is allowed. Second, answers are *many-valued*: a probe may have any number of distinct outputs across classes, and the adversary may return a value that no live class produces. Third, the budget is *global and adversarial*, not per-probe and not stochastic; there is no distributional assumption anywhere in this paper.

*Remark 2* (Off-table answers) An answer  $y_t$  that equals no live class output is not a special case. It disagrees with every live class simultaneously, so it is exactly the *catch-all outcome* of the partition  $\pi_{q_t}$ , and it is represented by a single **other** edge in every strategy we build. Corollary 23 shows the adversary never benefits from it, which is why our lower-bound certificates consist entirely of genuine catalog outputs and are therefore replayable transcripts.

## 2.3 Belief States Are a Sufficient Statistic

The residual game is determined by each class's number of disagreements with the transcript.

**Definition 3** (Robust belief state) Given a transcript, define the *mismatch counter*  $\mu(c) = |\{t : y_t \neq A(q_t, c)\}|$ . A class is *live* when  $\mu(c) \leq e$ . The *belief state* is the tuple  $S = (S_0, \dots, S_e)$  with  $S_k = \{c : \mu(c) = k\}$ ; its live set is  $L(S) = \bigcup_k S_k$ , and its *residual budget* for a live class is  $r(c) = e - \mu(c)$ . The initial state is  $S^{\text{init}} = (\mathcal{C}, \emptyset, \dots, \emptyset)$ .

**Proposition 4** (Sufficiency) Fix  $A$  and  $e$ . If two transcripts induce the same belief state, then the minimum worst-case number of further probes needed to identify the target is the same for both. Consequently  $\text{OPT}_A(e)$  is the value of the recursion

$$\text{OPT}(S) = \begin{cases} 0, & |L(S)| \leq 1, \\ 1 + \min_{\substack{q \in \mathcal{Q}: \\ q \text{ refines } L(S)}} \max_y \text{OPT}(S_{q,y}), & \text{otherwise,} \end{cases} \quad (1)$$

evaluated at  $S^{\text{init}}$ , where  $S_{q,y}$  is the successor state after answer  $y$  to probe  $q$  and the maximum ranges over the outputs of live classes together with the catch-all outcome when it leaves a class alive.

The successor operation admits a bitmask representation. Writing  $\beta \subseteq \mathcal{C}$  for the block of  $\pi_q$  consistent with  $y$  (so  $\beta = \emptyset$  for a catch-all answer),

$$(S_{q,y})_0 = S_0 \cap \beta, \quad (S_{q,y})_k = (S_k \cap \beta) \cup (S_{k-1} \setminus \beta) \quad (1 \leq k \leq e). \quad (2)$$

A state with no live class cannot arise from a valid transcript and is discarded. Note that OPT in (1) carries no depth parameter; that this is legitimate is the content of Lemma 21.

## 2.4 A Worked Example

The following three-class table admits a certificate valid for every error budget.

*Example 1* (A blind triple) Let  $\mathcal{C} = \{a, b, c\}$  and let  $\mathcal{Q} = \{q_1, q_2, q_3\}$  with

$$A(q_1, \cdot) = (0, 1, 1), \quad A(q_2, \cdot) = (0, 0, 1), \quad A(q_3, \cdot) = (1, 0, 1)$$

in the order  $(a, b, c)$ . Every pair is separated ( $q_1$  separates  $a$  from  $b$ ,  $q_2$  separates  $b$  from  $c$ ,  $q_3$  separates  $a$  from  $b$  and  $b$  from  $c$ ), yet no probe has three distinct outputs on  $\{a, b, c\}$ : each partition has exactly two blocks. At  $e = 0$  two probes suffice and one does not, so  $\text{OPT}_A(0) = 2$ . For general  $e$ , Theorem 13 gives  $\text{OPT}_A(e) \geq 3e + 2$  and Theorem 9 applied to the depth-two noiseless tree gives  $\text{OPT}_A(e) \leq 3e + 2$ , so

$$\text{OPT}_A(e) = 3e + 2 \quad \text{for every } e \geq 0.$$

The proof consists of a five-node strategy skeleton plus three class indices; it does not grow with  $e$ . The two depth-two program-derived laws in our evaluation use such witnesses: the scalar table on  $(0, 1, 5)$  and the sequence table on  $(2, 21, 38)$ . This is why they admit laws evaluated at  $e = 10^6$ .

Every probe, restricted to the triple, isolates one class from the other two. If the adversary is charged, it is charged on one class or on two; the learner can never split the ambiguity three ways. This is an *incidence* property of the probe system, invisible to any bound that reads the instance only through its class count and fan-out, which is precisely why the counting bounds of Section 7 lose ground as  $e$  grows, while the multicover bounds do not.

Table 1 summarises the notation used below.

## 3 Exact Query Learning

### 3.1 Exact Active Learning from Corrupted Queries

In query learning, a learner chooses inputs and uses their labels to identify a target concept [23]. For a finite Boolean hypothesis class  $\mathcal{H} \subseteq \{0, 1\}^{\mathcal{Q}}$ , asking for  $h^*(q)$  is a membership query. For functions with other output types, the same operation is a value query. Allowing an adversary to replace at most  $e$  responses yields precisely Definition 1, after merging hypotheses with identical rows. Thus

$$\text{minimum worst-case queries for exact learning on } \mathcal{Q} = \text{OPT}_A(e).$$

**Table 1** Notation. Objects above the rule are inputs; below it are derived quantities.

| Symbol                                         | Meaning                                                                               |
|------------------------------------------------|---------------------------------------------------------------------------------------|
| $\mathcal{H}, N$                               | catalog of executable hypotheses and its size                                         |
| $\mathcal{Q}, Q$                               | declared ordered probe alphabet and its size                                          |
| $\mathcal{C}, M$                               | semantic classes (behavioural equivalence classes on $\mathcal{Q}$ ) and their number |
| $A(q, c)$                                      | output of class $c$ on probe $q$ ; the behaviour table                                |
| $e$                                            | budget of adversarially corrupted answers                                             |
| $\pi_q, \pi_q _L$                              | class partition induced by $q$ , and its restriction to $L$                           |
| $\mu(c), r(c)$                                 | mismatch count and residual budget $e - \mu(c)$ of class $c$                          |
| $S = (S_0, \dots, S_e), L(S)$                  | belief state bucketed by mismatch count; its live set                                 |
| $\text{OPT}_A(e), \text{OPT}_A^{\text{na}}(e)$ | optimal adaptive depth; optimal non-adaptive size                                     |
| $d$                                            | depth of a verified noiseless ( $e = 0$ ) strategy                                    |
| $F(d, e) = (d + 1)e + d$                       | decreasing-residual robustification bound                                             |
| $B$                                            | bound on the number of outcome edges of any probe                                     |
| $V_d^B(r), W_B(S, d)$                          | Hamming ball volume and state character (Section 7)                                   |
| $\alpha e + \beta$                             | an affine all-budget law with slope $\alpha$ and intercept $\beta$                    |
| $\mathcal{I}, \Phi(L, r)$                      | isolation witness and its residual-game potential (Section 5.6)                       |

The equality follows by using the same query and stopping rule in the two descriptions: each valid labelled transcript is a valid game transcript, and conversely. A successful leaf names a semantic class. If the declared alphabet does not distinguish two global functions, the guarantee identifies their common behaviour on that alphabet only.

The mismatch profile is an error-tolerant version space. After a transcript  $\tau = ((q_1, y_1), \dots, (q_t, y_t))$ , the surviving hypotheses are

$$V_e(\tau) = \left\{ h \in \mathcal{H} : \sum_{i=1}^t \mathbb{1}\{h(q_i) \neq y_i\} \leq e \right\}.$$

The learner stops when this set contains one semantic class. Active query selection can reduce the worst-case cost because it responds to the remaining possibilities, whereas a non-adaptive learner must choose its complete query multiset before observing labels. Work on malicious membership-query answers [8] and simultaneous learning and covering with adversarial noise [24] provides direct learning-theoretic precedents for unreliable feedback.

The certificate adds a guarantee about the learning procedure’s *optimal cost*. Its upper component proves that a supplied learner always identifies the target within the claimed number of queries. Its lower component proves that every deterministic learner using the same alphabet can be forced to use that many. A separate checker can establish both statements without repeating strategy synthesis.

### 3.2 A Certified Concept-Learning Example

Consider all monotone conjunctions of four Boolean variables. Each hypothesis has the form  $h_S(x) = \bigwedge_{j \in S} x_j$ , for  $S \subseteq \{1, 2, 3, 4\}$ , with the empty conjunction equal to one. There are 16 hypotheses and the declared alphabet contains all 16 Boolean inputs. On

the input with only coordinate  $j$  equal to zero, the answer is zero exactly when  $j \in S$ . Four such queries therefore identify the concept in the absence of errors.

Proposition 20 proves that with at most two corrupted labels the optimal adaptive cost is 14 queries, while every non-adaptive design requires at least 20. The adaptive upper guarantee uses the known decreasing-residual robustification of a depth-four tree. A five-concept isolation witness supplies the matching lower guarantee. The supplied certificate for the full table occupies 2,795 bytes and proves the formula  $\text{OPT}_A(e) = 5e + 4$  for every nonnegative error budget. The supplementary learning example verifies this object and executes the certified learner on corrupted answers.

## 4 A Proof-Carrying Architecture

### 4.1 Producer, Verifier, and Trust Base

The *producer* may use exhaustive minimax search, greedy heuristics, memoisation, partition collapsing, or exact rational linear programming (LP). It is not trusted. Its only obligation is to emit an explicit certificate alongside each numerical claim.

The *verifier* receives the hypothesis catalog, probe alphabet, budget, and candidate object. It re-derives the catalog digest, rebuilds the semantic classes, recomputes the relevant table entries, and checks the certificate’s structural conditions without searching for an alternative strategy or bound. Portable law objects refer only to class and probe indices. Expanded adversary moves additionally name concrete output values; the verifier checks those values and recomputes every claimed successor state.

Figure 1 summarises the information flow. The shared trust base consists of the catalog evaluation routines, the binding computation, and the verifier. Search heuristics and the exact LP solver used to strengthen dual witnesses remain outside it. Thus a solver bug can weaken a result or fail to produce a certificate, but it cannot make an invalid certificate pass. The output-independent claim applies to the table binding and portable law objects, not to the concrete reply labels in expanded certificates.

### 4.2 Output-Free Table Binding

A certificate is bound to the declared table. Since general hashable outputs have no canonical byte encoding, the binding records the partitions they induce.

**Definition 5** (Table binding) The *binding* of  $A$  is the tuple  $(\text{cat}, \text{beh}, M, Q)$ , where  $\text{cat}$  is the catalog digest,  $M$  and  $Q$  are the class and probe counts, and the *behaviour digest*  $\text{beh}$  uses the 256-bit Secure Hash Algorithm (SHA-256):

$$\text{beh} = \text{SHA-256}\left(\text{canon}(\text{cat}, M, Q, (\text{sig}(\pi_{q_1}), \dots, \text{sig}(\pi_{q_Q})))\right),$$

where  $\text{sig}(\pi_q)$  relabels the outputs of  $q$  by order of first appearance while scanning classes in index order, and  $\text{canon}$  is a deterministic key-sorted encoding.

The binding encodes the catalog, class and probe counts, and the ordered canonical partitions. Adding a probe changes  $Q$ ; merging classes changes  $M$ ; reorderings that alter the canonical partition sequence change the encoded data. Exchanging probes

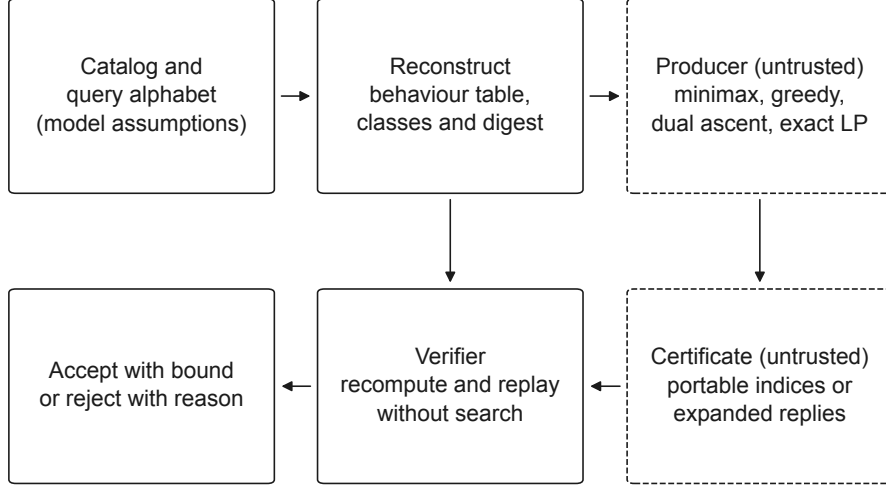


**Fig. 1** The proof-carrying pipeline. The catalog and alphabet are modelling assumptions; table reconstruction and the verifier are trusted. The producer and its supplied proof are untrusted. The verifier reconstructs the behaviour table from the catalog, checks the table binding, and replays the supplied object without consulting the search. Portable objects contain only indices, while expanded adversary moves also contain checked output labels

with identical partitions need not change the binding. Renaming an output value does not change a partition. A certificate carrying a binding that does not equal the verifier’s recomputed binding is rejected before any structural check runs.

### 4.3 Two Certificate Families

Table 2 compares the certificate families. Portable laws require a matching lower witness. The shallow rules have the completeness boundary of Proposition 15; isolation witnesses extend the interface to arbitrary basis depth (Theorem 18) without making it complete for all tables. The expanded family settles any cell whose search and proof construction finish, but its size tracks the belief graph.

## 5 Portable All-Budget Certificates

This section builds a proof object whose size does not depend on  $\epsilon$  and which, when it succeeds, settles the query complexity for every budget at once. The construction has three parts: an index-only description of a noiseless strategy, a robustification theorem that lifts it to any budget, and small lower witnesses that can match the lifted bound exactly.

### 5.1 Index-Only Noiseless Strategy Certificates

**Definition 6** (Indexed strategy certificate) An indexed strategy certificate consists of a binding (Definition 5), a root identifier, a claimed height, and a set of nodes. A node carries

**Table 2** The two certificate families.  $d$  is the depth of the noiseless basis strategy,  $M$  the class count,  $Q$  the alphabet size,  $k$  the witness size, and  $e$  the budget. “Nodes” is the number of nodes in the emitted proof object. Shared reconstruction includes catalog evaluation and  $O(M^2Q)$  semantic-class comparisons; see Appendix A.11.

|                    | Portable law certificate                                                                                     | Expanded optimality certificate                                                  |
|--------------------|--------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| Claim proved       | $\text{OPT}_A(e) = \alpha e + \beta$ for every $e \geq 0$ , or a verified $[L, U]$ gap for the shallow rules | $\text{OPT}_A(e) = k$ for one declared $e$                                       |
| Contents           | indexed noiseless tree; singleton or supplied $k$ -class isolation witness                                   | robust strategy tree; adversary DAG with pairwise or volume closures             |
| Size               | independent of $e$ ; $O(M)$ nodes                                                                            | tracks the reachable belief graph; observed up to $8.1 \times 10^5$ nodes        |
| Matching condition | singleton, or $k = d+1$ for a verified isolation witness (Thm. 18)                                           | any instance whose search and proof construction complete within declared limits |
| Verification       | shared table reconstruction; tree replay; $O(kQ)$ supplied-witness check                                     | $O(\text{nodes} \times Q \times M)$                                              |

an identifier, a *sorted* tuple of class indices, an optional probe index, and a tuple of child identifiers. A node with a single class index must be a leaf. For an internal node with class block  $L$  and probe index  $j$ , the children must correspond one-to-one and in order with the blocks of  $\pi_{q_j}|_L$ , ordered by their smallest class index.

Two design choices make the object output-free. Blocks are identified by their class membership rather than by the output value that produced them, and children are ordered canonically by smallest member. Consequently the payload contains only integers, and the verifier can regenerate the intended child blocks itself.

**Theorem 7** (Indexed strategy soundness) *If the verifier accepts an indexed strategy certificate with claimed height  $d$  against table  $A$ , then there is a noiseless ( $e = 0$ ) strategy for  $A$  that identifies every semantic class within  $d$  probes, and  $d$  is exactly the worst-case number of probes it uses.*

*Proof sketch.* The accepted conditions are: the binding matches; the root block is  $\{1, \dots, M\}$ ; every internal probe index is in range and refines its block; the declared children equal the recomputed canonical blocks of that restriction, both in number and in membership; singleton blocks are leaves; the node graph is acyclic with every node reachable from the root; and the recomputed height equals the claim. Reading the accepted object as a decision tree, following the block consistent with each observed output is well defined, and every leaf is a singleton, so the walk identifies the target. Since each internal node strictly refines, the walk terminates within the recomputed height. Full detail in Appendix A.  $\square$

## 5.2 Many-Valued Threshold Robustification

Let  $T$  be a verified noiseless strategy of depth  $d$ . The following procedure runs  $T$  against a corrupting oracle without ever materialising a robust belief state.

**Definition 8** (Decreasing-residual navigator) Maintain a current node of  $T$  and a residual budget  $r$ , initially  $e$ . At a node with declared outcome set  $Y$ , repeat that node's probe until some  $y \in Y$  has been observed  $r + 1$  times. Let  $\ell$  be the number of answers in the phase that differ from this winning outcome  $y$ , and descend along the  $y$ -edge with residual  $r - \ell$ . Answers outside  $Y$  count toward  $\ell$  but toward no outcome threshold.

A false outcome cannot reach the threshold because doing so would require  $r + 1$  corruptions. The winning outcome is therefore truthful, and every one of the  $\ell$  nonwinning answers is a corruption, including any off-table answer.

**Theorem 9** (Decreasing-residual robustification) *Let  $A$  be a finite many-valued behaviour table admitting a verified noiseless strategy of depth  $d \geq 1$ . Then for every  $e \geq 0$ ,*

$$\text{OPT}_A(e) \leq F(d, e) := (d + 1)e + d,$$

*and the bound is attained by the navigator of Definition 8. The navigator stores the current strategy node, its outgoing-label counters, and the residual budget, and never materialises the robust belief graph.*

*Proof sketch.* Induct on  $d$ . A phase at residual  $r$  costs  $r + 1 + \ell$  probes and hands the child residual  $r - \ell$ , so

$$r + 1 + \ell + F(d - 1, r - \ell) = F(d, r) - (d - 1)\ell \leq F(d, r).$$

The base case  $d = 0$  is a leaf at cost 0. □

The portable law object stores the noiseless strategy and lower witness, but no instantiated budget. Its verifier replays those objects and compares the resulting affine expressions symbolically, so verification does not grow with a budget at which the law is later evaluated. A separate fixed-budget threshold-strategy certificate instantiates the navigator at a declared  $e$ . Its verifier checks the recurrence for every  $1 \leq h \leq d$ ,  $0 \leq r \leq e$ , and  $0 \leq \ell \leq r$ , for a total of  $d(e + 1)(e + 2)/2$  integer inequalities. These are distinct certificate types.

*Remark 10* (Attribution) The numerical inequality and the decreasing-residual construction are, in the binary fault-tolerant decision-tree model, Lemma 2 of Maharjan and Watson [7], stated there as  $D_e(f) \leq D(f)(e + 1) + e$ . Here we use it in a finite many-valued restricted-query model with off-table answers and compose it with the budget-independent certificates and lower witnesses below.

Repeating every probe a fixed  $2e + 1$  times gives the bound  $(2e + 1)d$ . The difference from decreasing-residual repetition is

$$(2e + 1)d - ((d + 1)e + d) = e(d - 1),$$

because the decreasing schedule reduces the residual budget after observed corruptions. On our deepest primary instance ( $d = 9$ ) at  $e = 10^6$  this is a difference of 8,000,000 probes (Section 11.6). This is an algebraic comparison of two formulas, not a runtime measurement.

### 5.3 Compact Lower Witnesses

A matching lower bound must also be small. We first give the pair and blind-triple rules, then extend them to arbitrary-size isolation witnesses in Section 5.6.

**Lemma 11** (Pair lower law) *If  $A$  has two distinct classes  $c, c'$  separated by some declared probe, then  $\text{OPT}_A(e) \geq 2e + 1$  for every  $e \geq 0$ .*

*Proof sketch.* Restrict the game to  $\{c, c'\}$ ; by Lemma 22 this only lowers  $\text{OPT}$ . On any probe the adversary answers so as to charge whichever of the two classes currently has the larger residual budget, keeping the two mismatch counts balanced. After  $2e$  probes both classes still have  $\mu \leq e$ , so both remain live and the learner has not identified the target.  $\square$

**Definition 12** (Blind triple) A triple  $\{a, b, c\} \subseteq \mathcal{C}$  is *blind* if it is pairwise separated by  $\mathcal{Q}$  and no declared probe induces three distinct outputs on it, i.e.  $|\{A(q, a), A(q, b), A(q, c)\}| \leq 2$  for every  $q \in \mathcal{Q}$ .

**Theorem 13** (Blind-triple lower law) *If  $A$  contains a blind triple then  $\text{OPT}_A(e) \geq 3e + 2$  for every integer  $e \geq 0$ .*

*Proof sketch.* Consider a live blind triple with residual budgets  $r_a, r_b, r_c$ , and put  $S = r_a + r_b + r_c$ . We prove by induction on  $S$  that it needs at least  $S + 2$  further probes. When  $S = 0$ , no probe has three outcomes on the triple, so one probe cannot identify all three classes. For  $S > 0$ , any refining probe isolates one class, say  $c$ , from the other two. The adversary returns the output shared by  $a$  and  $b$ , thereby charging only  $c$ . If  $r_c > 0$ , all three remain live with residual sum  $S - 1$ , and induction gives a remaining cost of at least  $S + 1$ . If  $r_c = 0$ , class  $c$  dies but the live pair has residual sum  $S$  and needs at least  $S + 1$  probes by the pair argument. Thus every first probe has a successor requiring at least  $S + 1$  more probes, for a total of  $S + 2$ . Initially  $S = 3e$ , which gives the claim.  $\square$

The blind condition is checked against the full declared alphabet, not only the probes used by the upper strategy. A single additional probe that splits the three classes would invalidate the lower witness.

---

**Algorithm 1** Shallow compact-law discovery with explicit obstruction

---

**Require:** behaviour table  $A$  with  $M$  classes and  $Q$  probes

**Ensure:** an exact affine law with certificate, or a verified  $[L, U]$  gap with an obstruction category

- 1: compute binding and class output rows; **if**  $M = 1$  **return** law 0 with singleton witness
  - 2:  $j_1 \leftarrow$  a probe whose partition has  $M$  blocks, or  $\perp$  ▷ exhaustive:  $O(MQ)$
  - 3:  $P_2 \leftarrow$  a root probe all of whose blocks are made singleton by one further probe, or  $\perp$  ▷ exhaustive:  $O(MQ^2)$
  - 4: build noiseless tree  $T$ : use the shallow plan if  $j_1 \neq \perp$  or  $P_2 \neq \perp$ , else a deterministic greedy tree
  - 5:  $U \leftarrow F(\text{depth}(T), \cdot)$  ▷ Theorem 9
  - 6:  $\rho \leftarrow$  first separated pair;  $\tau \leftarrow$  first blind triple, or  $\perp$  ▷ exhaustive:  $O(M^3Q)$
  - 7: **if**  $j_1 \neq \perp$  **then**
  - 8:     **return** exact law  $2e + 1$  with pair witness  $\rho$  ▷ Corollary 14(2)
  - 9: **else if**  $P_2 \neq \perp$  **and**  $\tau \neq \perp$  **then**
  - 10:     **return** exact law  $3e + 2$  with blind-triple witness  $\tau$  ▷ Corollary 14(3)
  - 11: **end if**
  - 12:  $w \leftarrow \tau$  if  $\tau \neq \perp$  else  $\rho$ ;  $\lambda_0 \leftarrow 2$  if  $P_2 \neq \perp$  else 3 ▷ noiseless floor from the exhaustive searches
  - 13: **return** gap with upper  $U$ , lower  $L(e) = \max\{\lambda_0, \text{law}(w)(e)\}$ , and obstruction category  
      *missing blind triple* if  $P_2 \neq \perp$ , else *no depth-two strategy*
- 

## 5.4 Matched Laws

**Corollary 14** (Matched shallow laws) *Let  $A$  be a behaviour table.*

1. *If  $M = 1$  then  $\text{OPT}_A(e) = 0$  for all  $e$ .*
2. *If  $M \geq 2$  and some declared probe separates every pair of classes (a depth-one strategy exists), then  $\text{OPT}_A(e) = 2e + 1$  for all  $e$ .*
3. *If a strategy of depth two exists and  $A$  contains a blind triple, then  $\text{OPT}_A(e) = 3e + 2$  for all  $e$ .*

*Proof sketch.* Combine Theorem 9 at  $d \in \{0, 1, 2\}$  with the trivial bound, Lemma 11, and Theorem 13; in each case  $F(d, e)$  equals the lower law.  $\square$

Each case is a certificate whose upper and lower halves must agree exactly: the verifier recomputes the basis depth, requires the witness kind to match that depth (singleton with  $d = 0$ , pair with  $d = 1$ , blind triple with  $d = 2$ ), and rejects any claimed law that differs from  $F(d, e)$ . A certificate that pairs, say, a depth-two strategy with a pair witness is rejected even though both halves are individually valid, because their bounds do not meet.

## 5.5 Shallow Discovery and Its Obstructions

The three searches in Algorithm 1 are exhaustive over the explicit table.

**Proposition 15** (Completeness and Obstructions) *Algorithm 1 returns an exact law if and only if  $A$  falls in one of the three regimes of Corollary 14. When it returns a gap, both the upper and the lower bound are independently verifiable, and the obstruction category correctly records the outcome of the exhaustive shallow searches: missing blind triple means a depth-two strategy exists but no separated triple stays binary under the full alphabet; no depth-two strategy means no strategy of depth at most two exists, which certifies the noiseless floor  $\text{OPT}_A(0) \geq 3$ .*

The reported lower bound is  $L(e) = \max\{\lambda_0, \alpha e + \beta\}$ , where  $\lambda_0 \in \{0, 1, 2, 3\}$  is a floor established by exhaustive shallow search: if no depth- $\leq 2$  strategy exists then  $\text{OPT}_A(0) \geq 3$  regardless of any triple. At  $e = 0$  the floor binds. The blind-triple law contributes only 2, so reading the certificate as a bare affine expression would understate it. At  $e \geq 1$  the affine term dominates on all our instances.

The upper bound is always available, because a deterministic greedy noiseless tree always exists; the lower bound is what runs out of strength. On the point-function family the greedy tree has depth 9, so  $U(1) = 19$ , while the pair and blind-triple rules give only  $L(1) = 5$ , and the true value is 19. Section 5.6 closes this gap with a ten-class witness. The shallow obstruction remains valid; it does not exclude a deeper matched law.

## 5.6 Isolation Witnesses of Arbitrary Size

The pair and blind-triple arguments share a residual potential that extends to more classes. The uniform-budget singleton-search value is classical [21, 22]. Here we formulate the argument for unequal residual budgets and for a supplied witness inside an arbitrary table. The witness condition concerns the complete probe alphabet, including probes absent from the upper strategy.

**Definition 16** (Isolation witness) A set  $\mathcal{I} \subseteq \mathcal{C}$  of  $k \geq 2$  pairwise separated classes is an *isolation witness* if, for every  $q \in \mathcal{Q}$ , some block of  $\pi_q|_{\mathcal{I}}$  has size at least  $k - 1$ . Thus every probe is constant on  $\mathcal{I}$  or isolates one member from all the others.

This condition is hereditary: every subset with at least two members is also an isolation witness. A separated pair always qualifies, and a three-class isolation witness is exactly a blind triple. The number of witness classes is otherwise unrestricted.

**Theorem 17** (Exact residual value on an isolation witness) *Let  $L \subseteq \mathcal{I}$  be a live set, with nonnegative integer residual budgets  $r(c)$  for  $c \in L$ . In the restricted identification game its exact remaining value is*

$$\Phi(L, r) = \begin{cases} 0, & |L| \leq 1, \\ \sum_{c \in L} (r(c) + 1) - 1, & |L| \geq 2. \end{cases} \quad (3)$$

*Consequently a  $k$ -class isolation witness gives  $\text{OPT}_A(e) \geq k(e + 1) - 1$  for every  $e \geq 0$ .*

*Proof* For the upper bound, whenever at least two classes survive, ask a refining probe, which exists by pairwise separation. Every answer disagrees with at least one live class. Each disagreement removes one unit from  $\sum_c (r(c) + 1)$ : a positive residual decreases by one, and a zero-residual class is removed with its last unit. An answer outside the table charges every live class. A nonterminal child therefore has potential at most  $\Phi(L, r) - 1$ ; a terminal child has value zero. Repeating this argument gives a strategy of depth at most  $\Phi(L, r)$ . This upper argument requires separation but does not require isolation.

For the lower bound, consider any probe. If it is constant on the current live set, its truthful answer preserves the state. With at least three live classes and a refining probe, the isolation condition supplies an answer shared by all but one class. Return that answer. Only the exceptional class is charged. If its residual is positive it decreases by one; otherwise that class is removed. At least two classes remain in either case, and the potential decreases by exactly one.

With two live classes, if either residual is positive, answer a separating probe so that it charges a class with positive residual. Both survive and the potential again decreases by one. When both residuals are zero, the potential is one and at least one query is necessary. Induction on any proposed query horizon now shows that a strategy of depth less than  $\Phi(L, r)$  cannot succeed. Constant probes cause no difficulty in this induction because they consume a round while preserving the potential.

Every constructed finite transcript is consistent with a surviving class and that class's allowed residual budget. For a deterministic learner, fixing such a survivor as the target yields the same transcript along the chosen replies. Thus the adversary argument respects the fixed-target game of Definition 1. Finally, restrict the full game to  $\mathcal{I}$  and initialise every residual to  $e$ ; restriction cannot increase the optimum, giving the claimed full-table lower bound.  $\square$

The terminal clause in (3) matters: a single surviving class is identified even if its remaining error budget is large. The theorem covers unequal residual budgets directly, not only the equal-budget initial state.

**Theorem 18** (Matched isolation law) *Suppose  $A$  has a verified noiseless strategy of depth  $d \geq 1$  and an isolation witness of size  $d + 1$ . Then, for every integer  $e \geq 0$ ,*

$$\text{OPT}_A(e) = (d + 1)e + d. \quad (4)$$

*The noiseless strategy need not be assumed optimal: the matching witness proves its optimality at  $e = 0$  as well. For  $M = 1$ , the separately checked law is zero.*

*Proof* The lower bound from Theorem 17 at  $k = d + 1$  equals the upper bound from Theorem 9.  $\square$

*Example 2* (Adding a probe invalidates a lower witness) Four one-hot classes with only singleton probes form an isolation witness and have a depth-three noiseless tree, giving  $4e + 3$ . Add a probe that partitions them into two pairs. It violates the four-class isolation condition. The enlarged alphabet has a depth-two strategy and a blind triple, so its optimum is  $3e + 2$ . At one error, the value falls from seven to five. Even after rebinding the old object to the enlarged table, its lower witness must be rejected, although its original upper tree still succeeds.

---

**Algorithm 2** Isolation-law discovery for a supplied noiseless strategy

---

**Require:** table  $A$ , candidate indexed strategy  $T$ , positive subset cap  $C$

**Ensure:** a checked all-budget law, exhaustive absence at size  $d + 1$ , or a resource limit

- 1: independently check  $T$  and recompute its depth  $d$ ; reject invalid input
  - 2:  $k \leftarrow d + 1$ ; initialise the count of attempted candidate subsets to zero
  - 3: enumerate sorted subsets of  $\mathcal{C}$  of size  $k$  by depth-first extension
    - charge each attempted partial subset to the cap; return limit if exhausted
    - prune a partial subset if any declared probe violates isolation
  - 4: **if** a size- $k$  witness is found **then**
    - 5: construct the certificate and invoke the independent isolation verifier
    - 6: **return** the accepted law (zero for a singleton;  $ke + k - 1$  otherwise)
  - 7: **else**
    - 8: **return** absent for the requested witness size
  - 9: **end if**
- 

**Corollary 19** (Extremal separated tables) *For any separated table of  $M \geq 2$  classes,  $\text{OPT}_A(e) \leq M(e + 1) - 1$ . For each integer  $e \geq 0$ , equality holds if and only if the whole class set is an isolation witness.*

*Proof* The universal potential upper argument proves the inequality, and Theorem 17 proves equality for isolation tables. Otherwise some probe has every output block of size at most  $M - 2$ . For  $e = 0$ , asking it first leaves a separated block needing at most  $M - 3$  further queries. For  $e \geq 1$ , its first answer charges at least two classes, lowering the universal potential by at least two. Both cases give a total of at most  $M(e + 1) - 2$ . The case  $M = 2$  always satisfies the isolation condition.  $\square$

## 5.7 An Independently Checked Isolation Certificate

An isolation certificate contains a table binding, an indexed noiseless strategy, sorted witness class indices, and the claimed affine coefficients. It has no instantiated error budget and no expanded robust belief graph. The verifier rebuilds the table, checks the binding and distinct valid class indices, checks separation, and recomputes every restricted probe partition. It then invokes the indexed-strategy checker, recomputes depth  $d$ , requires  $k = d + 1$ , and checks the coefficients  $(k, k - 1)$ . The singleton case instead requires the depth-zero strategy, class index zero, and coefficients  $(0, 0)$ . Acceptance is sound by Theorem 18.

The supplied-witness check takes  $O(kQ)$  table-entry operations. Complete replay also includes catalog evaluation, the shared  $O(M^2Q)$  semantic-class checks, and upper-tree replay. These costs are polynomial in the explicit input and proof and independent of the error budget during symbolic verification. Discovery is outside the verifier.

Hereditary pruning in Algorithm 2 is sound: a partial set that fails isolation cannot become a witness by adding classes. The implementation defaults to one million attempted subsets. Its **gap** result means exhaustive absence of a matching witness for the supplied tree depth; **limit** makes no such assertion. Neither establishes that another exact affine law is impossible. Finding a witness can still take exponential

time. Known families can instead supply a constructed witness directly, followed by the same independent check.

The certified learner applies Definition 8 to the checked tree. Its state is a tree node, that node’s outcome counters, and the remaining error budget; off-table replies are counted as corruptions.

## 5.8 Family Consequences

**Proposition 20** (Three complete families) *For  $k \geq 2$ ,  $p \geq 1$ , and  $e \geq 0$ , the following adaptive and non-adaptive optima hold over the indicated complete probe alphabets.*

| <i>Family and probes</i>                                                         | $\text{OPT}_A(e)$ | $\text{OPT}_A^{\text{na}}(e)$ |
|----------------------------------------------------------------------------------|-------------------|-------------------------------|
| <i><math>k</math> point functions; all singleton probes</i>                      | $k(e+1) - 1$      | $k(e+1) - 1$                  |
| <i>All monotone conjunctions of <math>p</math> variables; all Boolean inputs</i> | $(p+1)e + p$      | $p(2e+1)$                     |
| <i>All <math>p</math>-bit strings; coordinate probes</i>                         | $(p+1)e + p$      | $p(2e+1)$                     |

*Proof* The point classes themselves form an isolation witness of size  $k$ , and scanning  $k - 1$  points gives the noiseless upper tree. For conjunctions, choose the conjunction of all  $p$  variables and the  $p$  conjunctions omitting one variable each. All answer one on the all-ones input. On an input with exactly one zero, only the conjunction omitting that variable answers one. With at least two zeros, all answer zero. Thus this is an isolation witness of size  $p + 1$ . Querying the  $p$  inputs with exactly one zero identifies any conjunction noiselessly. For binary strings, use the zero string and the  $p$  unit strings, together with the depth- $p$  coordinate-reading tree.

For the last two families, each of the  $p$  essential probes is the only probe separating a particular pair of classes. The distance criterion of Theorem 34 requires at least  $2e + 1$  copies of each such probe in any non-adaptive multiset. Repeating them all attains the resulting lower bound  $p(2e + 1)$ .

For point functions, let  $t_i$  be the multiplicity of singleton probe  $i$ . Every pair requires  $t_i + t_j \geq 2e + 1$ . One count equal to  $e$  and every other count equal to  $e + 1$  attains total  $k(e+1) - 1$ . If the minimum count is  $s \leq e$ , the total is at least  $s + (k-1)(2e+1-s) \geq k(e+1) - 1$ ; if  $s > e$ , the total is larger.  $\square$

For all Boolean functions on  $b$  input bits, the string result applies with  $p = 2^b$ .

The point-search formula is stated by Fraknói et al. [22, Section 2], with the one-error case proved by Meng et al. [21, Theorem 3(1) and Lemma 10]. In addition, Maharjan and Watson [7, Observation 3] already give the tight AND-function value  $D_e(\text{AND}_p) = p(e+1) + e$ . Section 13 compares these results with the isolation certificate construction.

## 6 Expanded Fixed-Budget Optimality Certificates

For a fixed budget, the expanded method searches for an exact value and emits matching upper and lower certificates. Well-foundedness permits a depth-free recursion, and dominance justifies the subset and partition reductions used below.

## 6.1 Two Structural Lemmas

**Lemma 21** (Well-foundedness) *Restrict attention to refining probes, i.e. those offering at least two distinct outputs among live classes. Under any answer to a refining probe, either a live class is removed or  $\sum_{c \in L} \mu(c)$  strictly increases. Hence the potential  $(|L|, e|L| - \sum_{c \in L} \mu(c))$  strictly decreases lexicographically, the reachable belief graph is a finite DAG, and the depth-free recursion (1) is well defined. Non-refining probes leave the state unchanged and are strictly wasteful.*

**Lemma 22** (Dominance) *Write  $S \preceq S'$  when  $L(S) \subseteq L(S')$  and  $\mu_S(c) \geq \mu_{S'}(c)$  for every  $c \in L(S)$ . Then  $\text{OPT}(S) \leq \text{OPT}(S')$ .*

*Proof sketch.* Any tree solving  $S'$  also solves  $S$ : mismatch counts are a starting value plus a path-determined increment, so  $L_S$  stays inside  $L_{S'}$  at every leaf, and a leaf with at most one  $S'$ -survivor has at most one  $S$ -survivor.  $\square$

Lemma 22 transfers lower bounds from a subset of classes to the full state. It also yields two corollaries that shrink the search and the proof.

**Corollary 23** (No Catch-All Is Needed) *The catch-all answer charges every live class simultaneously, whereas a genuine class output spares the classes agreeing with it. By Lemma 22 the child after a real answer dominates the child after the catch-all, so an adversary maximising OPT never needs it.*

Every answer named by a lower certificate is a genuine output of a semantic class. Its moves therefore produce a legal transcript against a candidate strategy.

**Corollary 24** (Partition sufficiency) *Game dynamics depend on a probe only through  $\pi_q$ , and at a given state only through  $\pi_q|_L$ . Probes may therefore be collapsed to distinct partitions before search and to distinct restricted partitions at each state, and certificates expanded back over the full declared alphabet afterwards.*

*Remark 25* (No symmetry reduction) The only reductions applied are the two partition collapses of Corollary 24. No automorphism or symmetry-group reduction is used anywhere in the implementation, including for the parity and all-Boolean-function tables.

## 6.2 The Solver

A belief state is stored as one class bitmask per mismatch count, so the transition (2) costs  $O(e)$  machine words rather than  $O(M)$  scalar steps. Algorithm 3 gives the search.

Memoisation uses depth-free keys, justified by Lemma 21. The internal lower bound LOWER uses only the pairwise and volume bounds of Section 7; the blind-triple, multicover, and protected-pair bounds are implemented and independently checkable but are *not* called inside the search; they can only be incorporated through the optional certified floor  $\phi$ . Reaching a declared resource limit produces a limit outcome rather than a numerical bound.

---

**Algorithm 3** Exact minimax depth by bounded DAG recursion

---

**Require:** behaviour table  $A$ , budget  $e$ , optional certified floor  $\phi$

**Ensure:** exact  $\text{OPT}_A(e)$ , or an explicit resource-limit outcome

```
1: collapse probes to distinct partitions; precompute block masks ordered by least
   member
2: start  $\leftarrow \max\{\text{LOWER}(S^{\text{init}}), \phi\}$ ; ceil  $\leftarrow \lfloor \text{GREEDYCOVER} \rfloor \cdot (2e + 1) \triangleright$  a repeated
   separating cover always works
3: for  $k = \text{start}$  to ceil do
4:   if  $\text{OPT}(S^{\text{init}}, k) \leq k$  then return  $k$ 
5:   end if
6: end for

7: function LOWER( $S$ )
8:   return  $\max\{1, \text{PAIRWISE}(S), \text{VOLUME}(S)\}$   $\triangleright$  Sections 7
9: end function

10: function OPT( $S$ , cap)  $\triangleright$  exact if  $\text{OPT}(S) \leq \text{cap}$ , else  $\text{cap} + 1$ 
11:   if  $|L(S)| \leq 1$  then return 0
12:   end if
13:    $\ell \leftarrow$  memoised LOWER( $S$ ); count the state; abort if the state cap is exceeded
14:   if  $\ell > \text{cap}$  then return  $\text{cap} + 1$ 
15:   end if
16:    $u \leftarrow$  memoised upper value for  $S$ ; if  $u \leq \text{cap}$  return  $u$ 
17:   best  $\leftarrow \min\{u, \text{cap} + 1\}$ 
18:   for all restricted partitions of  $L(S)$ , widest block first do
19:      $c \leftarrow \text{best} - 2$ ; if  $c < 0$  break
20:     evaluate children (hardest first); if any exceeds  $c$ , skip this probe
21:     best  $\leftarrow 1 + \max_{\text{children}}$ ; if best  $\leq \ell$  break
22:   end for
23:   memoise and return best
24: end function
```

---

### 6.3 Adversary DAGs with Analytic Closures

The upper direction is a robust strategy tree, verified exactly as in Theorem 7 but over belief states with an explicit **other** edge. The lower direction is the object that removes the need to repeat the search.

**Definition 26** (Adversary certificate) An adversary certificate carries a catalog digest, a probe digest, the budget, the declared probe list, a refuted depth  $k$ , a root, an optional root class subset, and nodes. A node carries a belief state, a remaining depth, and either (i) a *move* for *every* declared probe, a concrete output label and a child whose remaining depth is one less, or (ii) a *closure*: a pairwise closure naming two live classes, or a volume closure naming a live subset and a branching bound.

---

**Algorithm 4** Adversary DAG extraction

---

**Require:** solved game, refuted depth  $k$ , node cap

**Ensure:** a certificate refuting depth  $k$ , or an explicit limit outcome

```
1: function BUILD( $S, m$ )
2:   if ( $S, m$ ) already built return its identifier; abort if the node cap is reached
3:   if an analytic bound on a subset of  $L(S)$  exceeds  $m$  then
4:     return a node carrying that closure ▷ pairwise, else volume
5:   end if
6:   for all collapsed partitions  $\pi$  do
7:     pick the least-indexed live block whose child is still unsolvable in  $m - 1$ 
       probes
8:     record it as the reply for every declared probe with partition  $\pi$ 
9:   end for
10:  return a node whose moves recurse on BUILD( $\cdot, m - 1$ )
11: end function
```

---

Closures are what keep the object finite in practice: without them the DAG must unroll to remaining depth zero. With them it stops as soon as an analytic argument applies to a subset, which by Lemma 22 transfers to the node itself. Disabling closures is one of our ablations.

**Proposition 27** (Adversary-DAG soundness) *Suppose the verifier accepts an adversary certificate with refuted depth  $k$  against  $A$  and budget  $e$ . Then no strategy identifies the target within  $k$  probes, so  $\text{OPT}_A(e) > k$ . If in addition a strategy certificate of depth  $k + 1$  verifies, then  $\text{OPT}_A(e) = k + 1$ , and playing the certificate against the strategy yields an explicit transcript of legal answers on which at least two classes remain consistent.*

*Proof sketch.* Induct on remaining depth. A closed node carries a bound on a subset of its live classes exceeding its remaining depth; by Lemma 22 the node’s own optimum exceeds it too. An open node answers *every* declared probe with a legal label and lands on a valid node of remaining depth one less, so a solver can neither stop (two classes remain consistent) nor escape by choosing a different probe. Well-foundedness is immediate since remaining depth strictly decreases.  $\square$

The verifier requires an open node to answer every declared probe and recomputes each successor and depth decrement. It also checks that analytic closures establish the claimed lower bound and that every reachable node is well formed. These conditions are sufficient for the induction in Proposition 27.

*Remark 28* (Subset-rooted claims) A certificate may be rooted at a declared subset of at least two classes, all at mismatch zero. By Lemma 22 refuting depth  $k$  for the subset refutes it for the whole table, and the subset’s belief graph can be dramatically smaller. The verifier checks the subset is well formed and that the root state starts all its members at mismatch zero.

## 7 A Hierarchy of Checkable Lower Bounds

Lower bounds are where the difficulty concentrates, so we develop several, each reading a different feature of the instance. Table 3 summarises. The organising observation is that counting bounds see the probe system only through a scalar, its widest fan-out, while covering bounds see its incidence structure, and only the latter grow with  $e$  at the right rate.

### 7.1 Counting Bounds

The simplest is the *pairwise* bound: two live classes with residual budgets  $r, r'$  need  $r + r' + 1$  further probes (Lemma 11 generalised to a mid-game state), which in bitmask form is  $2e + 1 - k - k'$  for the two smallest occupied mismatch counts. The next is a Berlekamp-style volume argument, adapted to survive the two features that make the restricted-probe setting awkward: state-dependent non-uniform branching, and heterogeneous per-class budgets.

**Theorem 29** (Volume bound) *Let a belief state have live classes with residual budgets  $r(c)$ , and let  $B$  bound the number of outcome edges of every probe at every reachable descendant state. Put*

$$V_d^B(r) = \sum_{j=0}^{\min(r,d)} \binom{d}{j} (B-1)^j, \quad W_B(S, d) = \sum_{c \in L(S)} V_d^B(r(c)).$$

*If the state is solvable within  $d$  probes then  $W_B(S, d) \leq B^d$ .*

*Proof sketch.* Pad every leaf to depth  $d$  with virtual unary no-op steps, which leave the survivor set unchanged. Walk the resulting depth- $d$  tree from the root, choosing uniformly among the  $b_t \leq B$  edges at each real node and taking the sole edge at a virtual node. Fix a class  $c$ . While  $c$  has disagreed at most  $r(c)$  times it is live, so exactly one of the  $b_t$  real edges carries its own output and the walk disagrees with probability  $(b_t - 1)/b_t \leq (B - 1)/B$ . Virtual steps have disagreement probability zero. The disagreement count is therefore stochastically dominated by  $\text{Bin}(d, (B - 1)/B)$ , giving  $\Pr[c \text{ survives}] \geq V_d^B(r(c))/B^d$ . A solving tree leaves at most one survivor per leaf, so these events are pairwise disjoint and their probabilities sum to at most one. Multiply by  $B^d$ .  $\square$

Two features are worth noting. Non-uniform branching *helps*:  $b_t < B$  only lowers the disagreement probability, so the bound is valid with  $B$  any upper bound. And setting every residual to zero recovers  $M \leq B^d$ , so Theorem 29 strictly dominates the naive  $\lceil \log_B M \rceil$  counting bound. The implementation returns the least  $d$  not refuted, which is a genuine lower bound because  $W_B(S, d + 1) \leq B W_B(S, d)$  by the Pascal identity, making the predicate monotone in  $d$ .

**Corollary 30** (Subset volume) *Theorem 29 may be applied to any subset of the live classes, with  $B$  recomputed for that subset. Shrinking the subset lowers the character but also lowers  $B$ , which sits in the exponent; the strongest bound therefore comes from a large subset that every probe nearly fails to split.*

That trade-off is searched over greedily by growing, for each of a bounded range of branching caps and a bounded set of seed classes, a maximal subset on which no probe offers more outputs than the cap. Every candidate it returns is sound; the search budget affects only strength. Verification recomputes the subset's branching from the catalog and rejects a claimed branching smaller than the recomputed one, then checks the strict inequality  $W_B(S, d-1) > B^{d-1}$ .

## 7.2 Covering Bounds

Counting bounds must lose ground as  $e$  grows, for the reason Example 1 makes concrete: the binding structure is an incidence property. The next bound reads that structure directly.

**Theorem 31** (Truthful-path multicover) *Fix a belief state with mismatch counts  $\mu(\cdot)$ , a budget  $e$ , and a live reference class  $i$ . Suppose a strategy solves the state in  $d$  probes, and run it against the oracle that always answers the true output of  $i$ . Then  $\mu(i)$  never grows, so the run reaches a leaf that must identify  $i$ ; hence every other live  $j$  has accumulated  $d_j = e - \mu(j) + 1$  fresh mismatches, each necessarily at a probe separating  $j$  from  $i$ . Therefore the probes along that single path form a multiset covering each  $j$  at least  $d_j$  times, and*

$$\text{OPT}(S) \geq \max_{i \in L(S)} \text{MC}_i(S),$$

where  $\text{MC}_i$  is the minimum size of a probe multiset meeting the demands  $d_j$  under the incidence “ $q$  separates  $j$  from  $i$ ”.

Because  $\text{MC}_i$  is an integer set multicover, its LP relaxation has a dual that is trivially checkable: maximise  $\sum_j d_j y_j$  subject to  $\sum_{j \text{ sep. by } q} y_j \leq 1$  for every declared probe,  $y \geq 0$ . Any feasible  $y$  certifies  $\text{OPT} \geq \lceil \sum_j d_j y_j \rceil$ , and a verifier need only recompute the separation incidence from the catalog and check  $Q$  linear inequalities in exact rational arithmetic.

**Corollary 32** (Uniform special case) *If  $J$  is a set of live classes such that no probe separates more than  $t$  members of  $J$  from  $i$ , then  $y \equiv 1/t$  on  $J$  is feasible and  $\text{OPT} \geq \lceil (\sum_{j \in J} d_j)/t \rceil$ , which at the initial state is  $\lceil (e+1)|J|/t \rceil$ , linear in the budget, with slope  $|J|/t$  determined purely by how badly the probe system covers  $J$  from  $i$ .*

Corollary 32 is the first bound in the hierarchy whose growth rate in  $e$  matches that of  $\text{OPT}$ ; the counting bounds do not, and their measured shortfall on the scalar table grows as 0, 1, 3 for  $e = 0, 1, 2$ . The truthful-path construction does not use the reference class's remaining lie budget. The next bound follows a path that does.

**Theorem 33** (Protected pair) *Fix a state, a live reference  $i$  and a live protégé  $j$ , and let  $\Lambda = e - \mu(i)$ . Consider the oracle that answers the output of  $j$  whenever the probe separates  $i$  from  $j$  and fewer than  $\Lambda$  such answers have been given, and the output of  $i$  otherwise. Then  $i$  remains live throughout the run. Decomposing the path into lying and truthful steps gives a covering program: truthful steps separating  $i$  from  $j$  must number at least  $d_j$ ; every other*

**Table 3** The lower-bound hierarchy. “Reads” is what the bound sees of the probe system; “used in search” identifies bounds evaluated within Algorithm 3, rather than bounds supplied only through the optional certified floor  $\phi$ . Verification costs exclude the shared  $O(MQ)$  table rebuild.

| Bound                    | Reads                           | Value at $S^{\text{init}}$        | Verify           | Used in search |
|--------------------------|---------------------------------|-----------------------------------|------------------|----------------|
| Pairwise (Lem. 11)       | one separated pair              | $2e + 1$                          | $O(Q)$           | yes            |
| Volume (Thm. 29)         | class count, fan-out $B$        | least $d$ with $W \leq B^d$       | $O(MQ + d)$      | yes            |
| Subset volume (Cor. 30)  | a low-branching subset          | as above on the subset            | $O(MQ + d)$      | yes            |
| Blind triple (Thm. 13)   | a triple no probe splits        | $3e + 2$                          | $O(Q)$           | no             |
| Multicover (Thm. 31)     | separation incidence from $i$   | $\lceil \sum_j d_{ij} y_j \rceil$ | $O(MQ)$ rational | no             |
| Protected pair (Thm. 33) | incidence from $i$ and from $j$ | optimum of the $P^+$ dual         | $O(MQ)$ rational | no             |

*live  $k$  must be charged  $d_k$  times by truthful steps separating  $k$  from  $i$  or lying steps separating  $k$  from  $j$ ; and the path contains  $\Lambda$  lying steps before it can terminate. The optimum of this covering program is therefore a lower bound on the run length and on OPT.*

This covering program accounts explicitly for probes spent while the reference class’s residual budget is consumed, a feature absent from the truthful-path construction. Its certificate carries the two dual families as exact rationals. An exact rational simplex may strengthen the witness, but, as noted in Section 4, that solver is outside the trust base.

## 8 The Non-Adaptive Baseline

In the non-adaptive setting the learner commits to a probe multiset in advance, so its class behaviours form a code and the criterion is classical [25, 26].

**Theorem 34** (Distance criterion) *A probe multiset identifies every semantic class under at most  $e$  corrupted answers if and only if the induced class code has minimum Hamming distance at least  $2e + 1$ .*

*Proof sketch.* Sufficiency: radius- $e$  balls around distinct codewords are disjoint, so at most one class lies within distance  $e$  of any transcript, and nearest-codeword decoding returns it. Necessity: if classes  $i \neq j$  differ in only  $\delta \leq 2e$  coordinates, the transcript following  $j$  on  $\min(e, \delta)$  of those coordinates and  $i$  elsewhere is within  $e$  of both.  $\square$

The necessity direction is itself a certificate. The implementation emits a *confusion witness*, the explicit transcript together with both distances, and a verifier recomputes the code from the catalog and checks that both distances are at most  $e$ . A code that fails is thus refuted concretely, not merely by a distance count.

Theorem 34 turns the non-adaptive optimum into a purely combinatorial quantity: the minimum size of a probe multiset separating every class pair at least  $2e + 1$  times. This is a set multicover [27, 28], so it has an LP dual.

**Proposition 35** (Dual lower certificate) *Minimising  $\sum_q x_q$  subject to  $\sum_q a_{q,p} x_q \geq 2e + 1$  for every class pair  $p$  has dual: maximise  $(2e + 1) \sum_p y_p$  subject to  $\sum_{p \text{ split by } q} y_p \leq 1$  for every declared probe. Any feasible  $y$  certifies  $\text{OPT}_A^{\text{na}}(e) \geq \lceil (2e + 1) \sum_p y_p \rceil$ , and verification recomputes each probe's separation mask from the catalog and checks  $Q$  rational inequalities.*

The implementation closes the remaining gap by an iterative-deepening depth-first search over probes collapsed to distinct separation masks with dominated masks discarded, encoding the residual demand as level sets so that applying a probe is a fixed number of word operations, branching forcibly through the most deficient pair, and memoising refuted deficiency vectors. The scan starts at the dual bound. The reported size is therefore a proven optimum, and the reported adaptivity gap  $\text{OPT}_A^{\text{na}}(e) - \text{OPT}_A(e)$  is a difference of two certified quantities rather than of two heuristics.

## 9 Known Hardness at Zero Budget

At zero errors, finite restricted-query identification is the minimum-height decision-tree problem for an explicit object-test table. This objective was studied directly by Garey [12]. For binary tests, Laber and Nogueira [13] prove that no  $o(\log M)$ -approximation exists unless  $P = NP$ , and Bshouty and Makhoul [14] develop polynomial-time constructions and record the sharper  $(1 - o(1)) \ln M$  hardness threshold obtained using the Set Cover result of Dinur and Steurer [29]. The proposition below gives a self-contained reduction for the many-valued table model.

**Proposition 36** (Equality reduction at zero budget) *There is a polynomial-time map from any coverable Set Cover instance  $(U, \mathcal{S})$  to a behaviour table  $A$  with  $|U| + 1$  classes and  $|\mathcal{S}|$  probes such that  $\text{OPT}_A(0)$  equals the minimum cover size. Consequently computing  $\text{OPT}_A(e)$  exactly is NP-hard already at  $e = 0$ , and a polynomial-time exact algorithm would imply  $P = NP$ .*

*Proof sketch.* Introduce an *anchor* class answering 0 on every probe, and for each  $u \in U$  a class answering  $1 + u$  exactly on the probes whose set contains  $u$ , and 0 elsewhere; use one probe per set. Along the all-anchor transcript, any identifying strategy must ask probes whose sets cover  $U$ , since an element class agreeing with the anchor everywhere asked is indistinguishable from it. Conversely asking a minimum cover distinguishes every class from the anchor and, because element classes have distinct nonzero labels, from each other.  $\square$

The equality reduction is not gap-preserving, so it does not by itself establish an approximation threshold. This limitation of the reduction should not be confused with an absence of such results: logarithmic inapproximability is already known for

**Table 4** The fifteen primary instances.  $N$  is the hypothesis catalog size,  $M$  the number of semantic classes after behavioural reduction,  $Q$  the declared probe alphabet size,  $|\pi|$  the number of *distinct* induced partitions, and  $B$  the maximum probe arity. For random tables,  $s$  denotes the seed; the remaining family parameters are defined in Appendix B. Where  $N > M$  the semantic reduction compresses syntactic duplicates; where  $Q > |\pi|$  the alphabet contains interchangeable probes (Corollary 24).

| Setting                  | Family                | $N$ | $M$ | $Q$ | $ \pi $ | $B$ |
|--------------------------|-----------------------|-----|-----|-----|---------|-----|
| $r = 1$                  | Scalar DSL            | 30  | 16  | 9   | 9       | 5   |
| $r = 2$                  | Scalar DSL            | 30  | 16  | 25  | 17      | 8   |
| $d = 1$                  | Sequence DSL          | 9   | 9   | 220 | 66      | 9   |
| $d = 2$                  | Sequence DSL          | 73  | 49  | 220 | 210     | 40  |
| $(M, Q, s) = (6, 5, 11)$ | Binary random table   | 6   | 6   | 5   | 5       | 2   |
| $(M, Q, s) = (8, 7, 23)$ | Binary random table   | 8   | 8   | 7   | 7       | 2   |
| $(M, Q, s) = (7, 6, 31)$ | Ternary random table  | 7   | 7   | 6   | 6       | 3   |
| $n = 7$                  | Thresholds            | 8   | 8   | 7   | 7       | 2   |
| $n = 15$                 | Thresholds            | 16  | 16  | 15  | 15      | 2   |
| $k = 3$                  | Parities              | 8   | 8   | 7   | 7       | 2   |
| $k = 4$                  | Parities              | 16  | 16  | 15  | 15      | 2   |
| $n = 6$                  | Point functions       | 6   | 6   | 6   | 6       | 2   |
| $n = 10$                 | Point functions       | 10  | 10  | 10  | 10      | 2   |
| $b = 2$                  | All Boolean functions | 16  | 16  | 4   | 4       | 2   |
| $k = 3$                  | Monotone conjunctions | 8   | 8   | 8   | 8       | 2   |

the binary minimum-height problem [13, 14]. Proposition 36 also says nothing about certificate size. Our verification is polynomial in the explicit table and the supplied proof, and we make no claim that polynomial-size optimality proofs exist for all instances. An explicit family, the universe of non-empty subsets of  $\{1, \dots, n\}$  with set  $i$  containing the subsets containing  $i$ , exhibits a growing gap between the volume bound and the optimum. This witnesses a limitation of that bound and nothing more.

## 10 Experimental Design

The evaluation measures compact-certificate coverage, baseline optimality, the value of adaptivity, and synthesis and verification costs. It also records resource limits, solver ablations, and agreement between two execution environments.

### 10.1 Instances

Table 4 lists the fixed primary suite: fifteen deterministic instances across seven families, spanning program-derived tables with duplicate behaviours and classical concept classes with different probe-partition structures. Two are derived from a scalar straight-line program domain-specific language (DSL), two from a sequence-transformation DSL; the remainder are threshold, parity, point-function, all-Boolean-function, and monotone-conjunction tables plus three seeded random tables. Full construction detail is in Appendix B.

A fixed *extended sweep* contains 101 instances: 75 random tables, three fixed shapes (6 classes/5 probes/arity 2; 8/7/2; 7/6/3) crossed with every seed in  $\{0, \dots, 24\}$ , and 26

structured size sweeps: odd threshold domains  $3, 5, \dots, 19$ ; parity dimensions  $2, \dots, 5$ ; point counts  $3, \dots, 10$ ; monotone conjunctions on  $2, 3, 4$  variables; and all Boolean functions on 1 and 2 input bits. Its largest table has 32 classes and 31 probes. The sweep specification comprises these families, shapes, seeds, budgets, and resource caps.

The isolation evaluation rebuilds both suites without changing their declarations. An additional random evaluation crosses shapes  $(M, Q, B) = (6, 5, 2), (8, 7, 2), (7, 6, 3), (9, 8, 3)$  with seeds  $1000, \dots, 1063$ , giving 256 further tables. It uses the same synthetic generator and is reported separately from the original sweep. Three larger constructed certificates cover 64 point functions, six-variable monotone conjunctions, and all Boolean functions on three input bits (256 classes).

## 10.2 Protocol

Each *cell* is an (instance, budget) pair. The original fixed-budget evaluation runs the same six methods on every cell: shallow compact discovery (Algorithm 1); fixed per-node repetition  $(2e + 1)d$ ; decreasing-residual repetition  $F(d, e)$ ; robust greedy synthesis; exact two-sided adaptive certification (Algorithms 3 and 4); and exact non-adaptive optimisation (Section 8).

A numerical result is recorded after the corresponding evidence passes its independent verifier. A run that exceeds its cap records the limiting stage and no value for that method. Coverage rates include these cells; comparisons requiring a certified optimum use only completed cells.

Caps for the primary suite are 250k greedy states, 500k exact states, 500k adversary nodes, and 500k non-adaptive nodes; the sweep uses 1M greedy and 2M for the other three, and the  $e = 2$  phase adds a 600-second external wall cap per cell. A supplementary 13-cell  $e = 2$  suite adds targeted small-instance coverage, including two DSL tables, under the same method caps; the depth-two sequence table has a specified 1800-second wall-clock cap.

The isolation extension supplies Algorithm 2 with the same checked noiseless tree used by the shallow producer, under a one-million-subset cap. Every emitted object is serialised, parsed, and independently verified. We compare accepted laws with the frozen exact results and retain all gaps and limits. The larger structured experiment constructs witnesses from Proposition 20; it measures checked construction, not generic subset discovery.

## 10.3 Timing Protocol

Combinatorial outputs are deterministic for a fixed instance and budget. Primary timings are a single wall-clock observation per cell on CPython 3.14.5, arm64 macOS. The repeated fixed-budget study adds, for each of the 30 primary cells, one discarded warm-up and five serially executed recorded repetitions on CPython 3.12.3, x86-64 Linux, from which we report medians and observed ranges. A timing repetition is retained only when all non-timing outputs agree with the reference run. We report no confidence intervals, do not randomise execution order, and do not compare runtimes across the two machines.

**Table 5** Two program-derived tables with shallow compact laws over their complete declared alphabet. Payload is the serialised size of the portable certificate; discovery and verification are wall-clock seconds. The law is evaluated symbolically, so verification is independent of  $e$ ; evaluating the law incurs integer arithmetic whose bit cost grows with the representation of  $e$ .

| Table             | $M$ | $Q$ | blind<br>triple | basis<br>$d$ | payload<br>(B) | verify<br>(s) | OPT( $10^6$ ) |
|-------------------|-----|-----|-----------------|--------------|----------------|---------------|---------------|
| Scalar, radius 2  | 16  | 25  | (0, 1, 5)       | 2            | 4,591          | 0.0006        | 3,000,002     |
| Sequence, depth 2 | 49  | 220 | (2, 21, 38)     | 2            | 10,385         | 0.0479        | 3,000,002     |

Isolation-certificate timings use CPython 3.14.5 on arm64 macOS 26.6.2, with one warm-up and seven serial recorded replays. Payload sizes count the exact compact JavaScript Object Notation (JSON) bytes saved and hashed. These measurements are reported separately from earlier expanded-proof timings; we infer no speed-up ratio from measurements collected under different conditions.

Figures were generated from the fixed evidence with Matplotlib 3.10.9; the generation source is included in Online Resource 1.

## 10.4 Use of Generative AI

OpenAI Codex assisted with literature review, development and checking of mathematical arguments and code, and manuscript preparation. The author is responsible for the manuscript’s claims, references, and results.

## 11 Results

The original evaluation completes all methods on all 30 primary cells and all 202 sweep cells at  $e \in \{0, 1\}$ . At  $e = 2$ , expanded two-sided proofs complete on 98 of 101 sweep cells, greedy strategies on all 101, and exact non-adaptive codes on 100. Isolation certificates independently settle the two point-function proof limits, giving 100 exact adaptive values at  $e = 2$  and 302 across the 303 declared sweep cells. The remaining adaptive case is parity dimension five. Original method-level limits remain recorded in Section 11.6; they are not relabelled as completed searches.

### 11.1 Shallow-Rule Coverage

Shallow compact equality closes on 4 of 15 primary instances (8 of 30 cells) and 25 of 101 sweep instances, at every budget. The scalar radius-two, sequence depth-two, and random ternary tables certify  $\text{OPT}(e) = 3e + 2$  via a blind triple and a depth-two basis; the sequence depth-one table certifies  $\text{OPT}(e) = 2e + 1$  via a separating probe. The remaining 11 primary instances have *no* strategy of depth at most two, so Algorithm 1 returns a verified gap with this obstruction. The complementary case, in which a depth-two strategy exists but no blind triple is available, does not occur in these instances.

**Table 6** Exact formulas valid for every error budget. No subset-search limit was reached in these evaluations. Populations are kept separate because their families and some instances overlap. A remaining gap concerns matching the supplied basis, not the nonexistence of any compact proof.

| Population                   | Tables | Shallow laws | Isolation laws | Remaining gaps |
|------------------------------|--------|--------------|----------------|----------------|
| Original primary suite       | 15     | 4            | 9              | 6              |
| Original extended sweep      | 101    | 25           | 69             | 32             |
| Additional random evaluation | 256    | 71           | 161            | 95             |

Table 5 records the two program-derived laws. The claims are sharply table-specific, and one control makes this concrete: shrinking the scalar probe radius from 2 to 1, which removes 16 of 25 probes, destroys the depth-two strategy. That variant falls back to a verified gap, and its exact value at  $e = 1$  is 6, strictly between the compact bounds 5 and 7. A law is a statement about a table and its alphabet, not about a DSL.

The shallow compact *lower* bound is much weaker than the compact upper bound. It equals OPT in 14 of 30 primary cells, but the split by budget is revealing: 10 of 15 at  $e = 0$ , where the exhaustive noiseless floor  $\lambda_0$  is doing the work, and only 4 of 15 at  $e = 1$ , where the affine witness must carry the bound alone. In the sweep the same pattern is sharper: 83 of 101 at  $e = 0$  against 25 of 101 at  $e = 1$  and 25 of 98 at  $e = 2$ , exactly the cells where a matched law exists. The worst case is the 10-class point-function table at  $e = 1$ : the blind-triple witness gives 5 while the optimum is 19. The witness remains sound but is not informative in this cell; Figure 2b shows the distribution.

## 11.2 Isolation-Witness Coverage

The isolation rule proves exact all-budget laws on 69 of the 101 sweep tables, compared with 25 for the shallow rules. It retains every shallow success. Coverage in the primary suite rises from 4 to 9 of 15. These gains use the same noiseless basis strategies; they come from stronger checked lower witnesses rather than improved upper-tree search.

Table 6 reports the coverage of the all-budget laws.

Within the sweep, random-table coverage rises from 20 to 54 of 75, point functions from 1 to all 8, and Boolean-function families from 2 to all 5. Parity and thresholds retain one compact success each, out of four and nine tables respectively. The 69 formulas agree with all 205 overlapping previously certified fixed-budget values. Two additional values settle the nine- and ten-point instances at  $e = 2$ , formerly limited by expanded proof size. Combined with the frozen results, they give 302 exact cells out of 303.

Table 7 records certificate sizes and independent replay times.

The additional random evaluation gives 161 exact laws against 71 for the shallow rules. All 90 new successes come from the binary shapes: 47 of 64 at  $(6, 5, 2)$  and 43 of 64 at  $(8, 7, 2)$ , where the shallow rule has none. Both methods prove 53 of 64 laws at  $(7, 6, 3)$  and 18 of 64 at  $(9, 8, 3)$ . Thus the improvement depends on the table’s partition structure. The seeds differ from those of the initial sweep; both evaluations use the same synthetic generator.

**Table 7** Complete isolation-certificate payloads and independent replay. Sizes count the saved compact JSON bytes, including the table binding, indexed tree, witness indices, and affine claim. The table and catalog are supplied separately to the checker. Timings are medians of seven serial replays after one warm-up on arm64 macOS with CPython 3.14.5.

| Table                               | All-budget law | Bytes | Replay (ms) |
|-------------------------------------|----------------|-------|-------------|
| Nine point functions                | $9e + 8$       | 1,813 | 0.188       |
| Ten point functions                 | $10e + 9$      | 1,970 | 0.226       |
| Four-variable monotone conjunctions | $5e + 4$       | 2,795 | 0.498       |
| All two-input Boolean functions     | $5e + 4$       | 2,787 | 0.244       |

**Table 8** Larger explicit tables with constructed and independently checked isolation certificates. Median replay times are 11.166, 10.376, and 30.962 milliseconds, respectively, under the same seven-replay protocol.

| Constructed table                  | $M$ | $Q$ | Exact law  | Payload (B) |
|------------------------------------|-----|-----|------------|-------------|
| Point functions                    | 64  | 64  | $64e + 63$ | 15,389      |
| Six-variable monotone conjunctions | 64  | 64  | $7e + 6$   | 10,135      |
| All three-input Boolean functions  | 256 | 8   | $9e + 8$   | 42,380      |

### *Constructed larger certificates.*

The witnesses of Proposition 20 can be supplied directly. Table 8 validates three such constructions against their full tables. These are not measurements of generic witness discovery. The 256-class Boolean certificate contains 511 noiseless-tree nodes and nine witness indices and proves  $\text{OPT}(e) = 9e + 8$ .

Independent residual-game computations agree on 10,912 heterogeneous profiles and 381 four-class alphabet/budget comparisons. The additional random evaluation contains 240 reference game values, including 165 comparisons with accepted isolation laws, with no disagreement. Online Resource 1 provides the validation procedures, tests, and reproduction commands.

## 11.3 Robustification and Greedy Baselines

The decreasing-residual bound equals  $\text{OPT}$  in 25 of 30 primary cells: all 15 at  $e = 0$ , where  $F(d, 0) = d$ , and 10 of 15 at  $e = 1$ . Table 9 lists the five failures.

In the sweep, robustification exceeds the optimum in 27 of 101 cells at  $e = 1$  and in 30 of 100 certified cells at  $e = 2$ . The failures concentrate by family (Table 10). For threshold tables, robustification is tight in 2 of 9 at  $e = 1$  and 1 of 9 at  $e = 2$ . Point-function tables show the opposite behaviour: robustification is tight in all of them because the greedy noiseless tree is forced to be a linear scan.

Greedy synthesis is optimal in 26 of 30 primary cells (11 of 15 at  $e = 1$ ) and 76 of 101 sweep cells at  $e = 1$ . On parity with  $k = 4$  at  $e = 1$  it returns 8 where the optimum is 7, and on three other cells it reproduces the non-tight robustification value exactly.

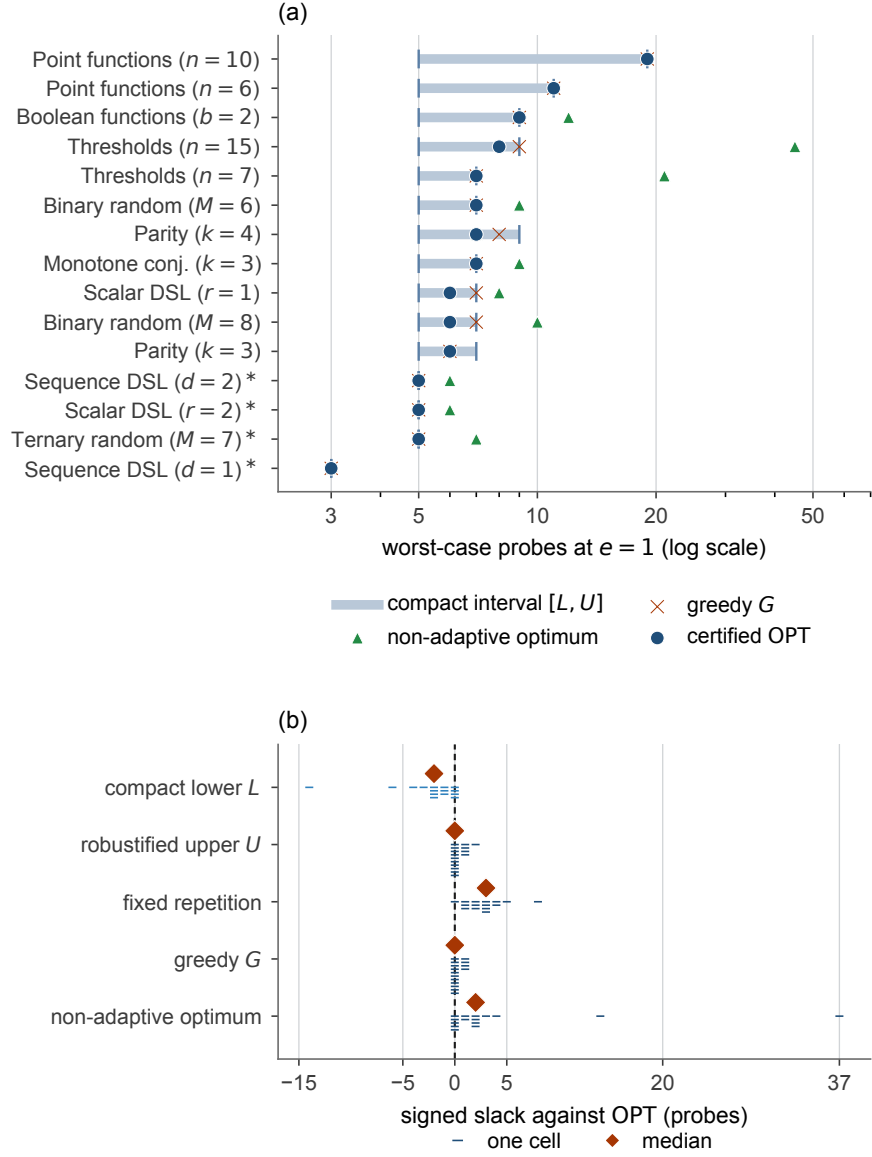


**Fig. 2** Original shallow-rule baseline. (a) Every primary instance at  $e = 1$ , sorted by certified optimum. The bar spans the shallow compact bounds  $[L, U]$ ; markers give the certified OPT, greedy depth, and exact non-adaptive size. Instances marked  $*$  admit a matched shallow law, so  $L = U = \text{OPT}$ . (b) Signed slack of each method against OPT across the same 15 cells. Each short tick denotes one cell; diamonds mark medians. The compact upper and greedy bounds sit close to zero; the compact lower bound and the non-adaptive baseline do not

**Table 9** Primary cells where the decreasing-residual upper bound is not tight. All occur at  $e = 1$ . Greedy synthesis matches the non-tight bound in three of the five.

| Instance                  | basis<br>$d$ | compact<br>$L$ | robust<br>$U = F(d, 1)$ | certified<br>OPT | greedy |
|---------------------------|--------------|----------------|-------------------------|------------------|--------|
| Scalar DSL ( $r = 1$ )    | 3            | 5              | 7                       | 6                | 7      |
| Binary random ( $M = 8$ ) | 3            | 5              | 7                       | 6                | 7      |
| Thresholds ( $n = 15$ )   | 4            | 5              | 9                       | 8                | 9      |
| Parity ( $k = 3$ )        | 3            | 5              | 7                       | 6                | 6      |
| Parity ( $k = 4$ )        | 4            | 5              | 9                       | 7                | 8      |

**Table 10** Per-family breakdown of the extended sweep after adding isolation certificates. Counts are out of cells with an exact adaptive certificate, shown as  $n$ . “Law” is a matched isolation all-budget law, “robust” and “greedy” are exact agreement with the certified optimum, and “adapt<sup>+</sup>” is a strictly positive certified adaptivity gap.

| $e = 1$        |     |     |        |        |                    |
|----------------|-----|-----|--------|--------|--------------------|
| Family         | $n$ | law | robust | greedy | adapt <sup>+</sup> |
| Random table   | 75  | 54  | 58     | 58     | 75                 |
| Threshold      | 9   | 1   | 2      | 2      | 9                  |
| Point function | 8   | 8   | 8      | 8      | 0                  |
| Parity         | 4   | 1   | 1      | 3      | 0                  |
| Boolean class  | 5   | 5   | 5      | 5      | 5                  |
| Total          | 101 | 69  | 74     | 76     | 89                 |
| $e = 2$        |     |     |        |        |                    |
| Family         | $n$ | law | robust | greedy | adapt <sup>+</sup> |
| Random table   | 75  | 54  | 55     | 55     | 75                 |
| Threshold      | 9   | 1   | 1      | 1      | 9                  |
| Point function | 8   | 8   | 8      | 8      | 0                  |
| Parity         | 3   | 1   | 1      | 2      | 2                  |
| Boolean class  | 5   | 5   | 5      | 5      | 5                  |
| Total          | 100 | 69  | 70     | 71     | 91                 |

Fixed per-node repetition is exact in 16 of 30 primary cells against 25 of 30 for the decreasing schedule, and at  $e = 2$  in the sweep it is exact in *zero* of 100 certified cells. Their algebraic gap  $e(d - 1)$  accounts for the full difference.

## 11.4 Certified Adaptivity Gaps

Figure 3 compares the methods and the two query regimes. Because both quantities are certified, the gap  $\text{OPT}_A^{\text{na}}(e) - \text{OPT}_A(e)$  is a difference of proven optima. It is strictly positive in 15 of 30 primary cells (10 of 15 at  $e = 1$ ), 89 of 101 sweep cells at  $e = 1$ , and 91 of 100 certified sweep cells at  $e = 2$ . The extremes are large: for thresholds on

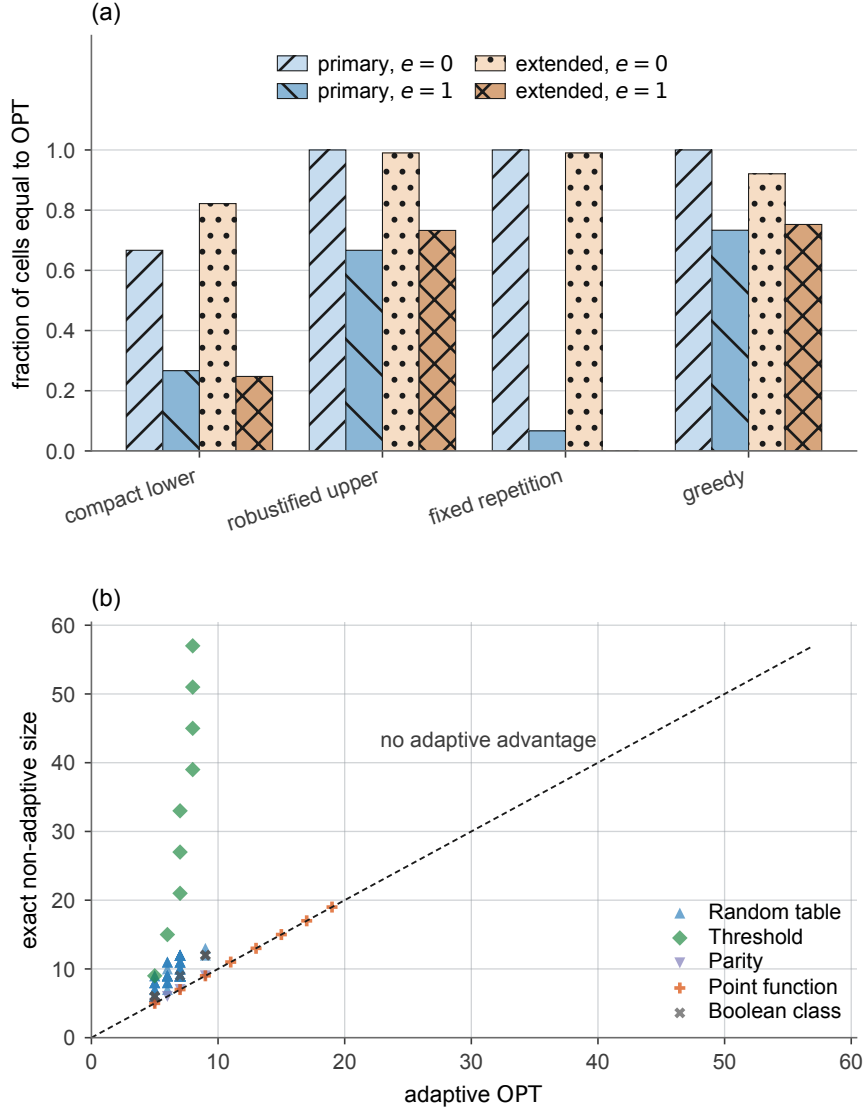


**Fig. 3** Original shallow-rule comparisons and certified adaptivity. (a) Fraction of cells in which each baseline equals the certified optimum in the original completed runs, by suite and budget, distinguished by hatch patterns. Fixed repetition collapses at  $e = 1$ ; the shallow compact lower bound collapses with it. (b) Certified non-adaptive optimum against certified adaptive optimum on the 101 sweep instances at  $e = 1$ . Points on the diagonal are instances where adaptivity provably does not help; the point-function and parity families sit there, while threshold tables depart from it sharply

[15] at  $e = 1$  the adaptive optimum is 8 while the shortest valid non-adaptive code has length 45, a gap of 37; on the 20-class threshold table in the sweep the gap is 49 at  $e = 1$  (8 versus 57) and 84 at  $e = 2$  (11 versus 95).

Every point-function cell has gap zero at both budgets, and every certified parity cell has gap zero at  $e = 1$  (Table 10). For point functions the reason is structural: the only available probes are singleton tests, and the pair-distance constraints attain the same value as the adaptive isolation law (Proposition 20).

### 11.5 Certificate Size and Verification Cost

Figure 4 relates proof size to synthesis and replay cost. In the reference primary run, replay is faster than synthesis in 28 of 30 cells. The repeated experiment sharpens this to 29 of 30 by median of five runs. The largest ratio is on the depth-two sequence table at  $e = 1$ : median synthesis 8.839 s (observed range 8.779–8.872) against median replay 0.231 s (0.230–0.233), a factor of 38.2. The sole reversal is the tiny noiseless depth-one sequence cell (0.0042 s synthesis, 0.0057 s replay), where fixed replay overhead, rebuilding a 220-probe table, dominates a trivial search.

Checking cost depends on certificate size. On the 10-class point-function table at  $e = 1$ , 59,028 search states produce a 380,112-node adversary DAG with 3,218,910 moves, and median synthesis and replay are 22.198 s and 6.742 s: a ratio of only 3.3. The sweep produces something larger still: for monotone conjunctions on 4 variables at  $e = 2$ , 304,918 search states yield an *812,971-node* DAG with 8,911,248 moves, proving  $\text{OPT}(2) = 14$ . The isolation certificate for this same conjunction table instead occupies 2,795 bytes and applies to every budget. The expanded format still has no general small-proof guarantee.

Portable certificates behave differently because their size follows the noiseless tree and supplied witness. Table 11 records the original shallow-rule scaling comparison, including a 256-class Boolean-function gap object. The new isolation object for that table has 511 tree nodes, nine witness indices, and 42,380 bytes; it now proves the exact law  $9e + 8$  (Table 8). These are finite explicit tables, and the timings of the two evaluations are not used for a speed-up comparison.

### 11.6 Results at Error Budget Two

Figure 5 shows the budget dependence of certified depths and repetition schedules. The original full sweep at  $e = 2$  records an outcome in all 101 cells without an external timeout or process failure. Exact two-sided proofs complete in 98, greedy in all 101, and exact non-adaptive optimisation in 100. All 75 random-table proofs complete. Among the 98 expanded completions, shallow laws close 25, robustification is tight in 68, greedy is optimal in 69, and 91 have a strictly positive adaptivity gap.

Four method-level limits occur:

- parity with  $k = 5$  exceeds the two-million exact-state cap;
- point-function tables with  $n = 9$  and  $n = 10$  each exceed the two-million adversary-certificate node cap, so proof construction rather than search reaches the limit;
- parity with  $k = 5$  separately exceeds the two-million node cap in non-adaptive optimisation.

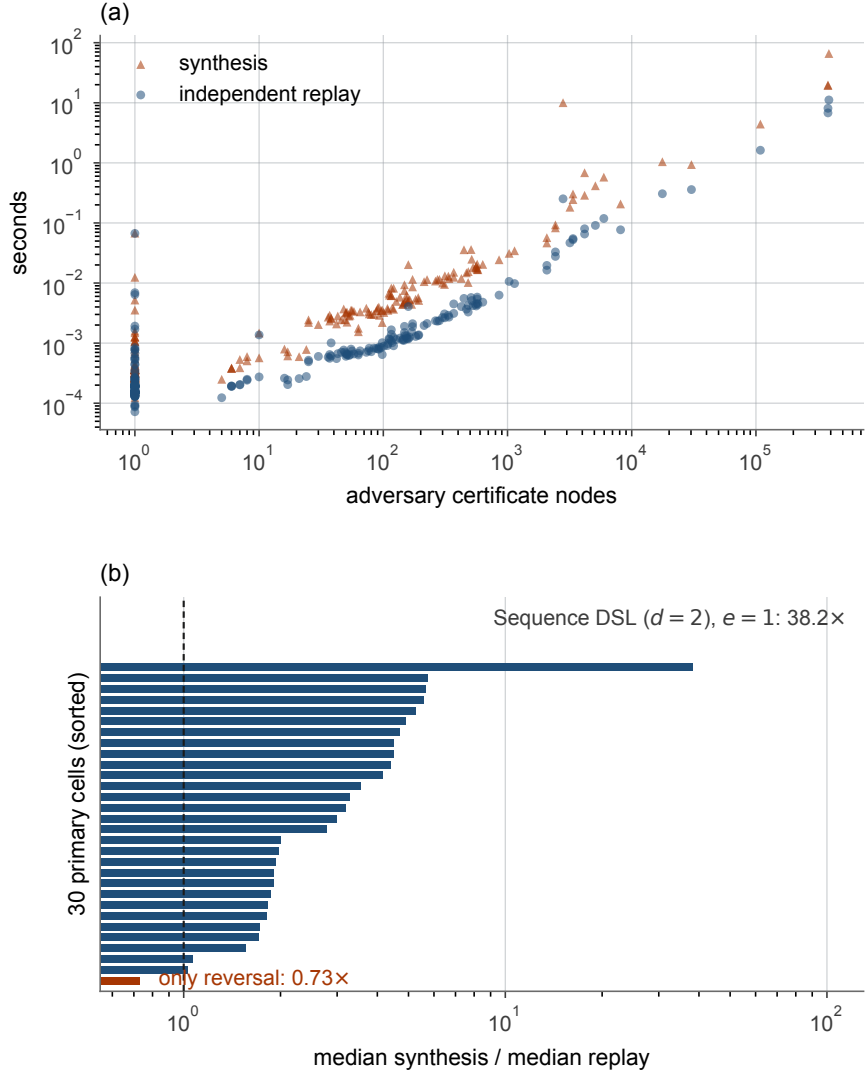


**Fig. 4** (a) Synthesis and independent replay time against adversary certificate size across 232 cells (log-log); triangles mark synthesis and circles mark replay. Replay lies below synthesis almost everywhere, but both grow with the proof, so cheap checking is a consequence of small proofs rather than a property of the format. (b) Median synthesis divided by median replay over five recorded repetitions on the 30 primary cells. The single reversal is the smallest cell, where fixed replay overhead dominates

No failed method is assigned a value. The parity event is a search-state limit, whereas the point-function events occur during certificate construction after search. Separate isolation certificates now prove  $\text{OPT}(2) = 26$  for nine points and  $\text{OPT}(2) = 29$  for ten points, using 1,813 and 1,970 bytes. Thus 100 of the 101 cells have certified adaptive

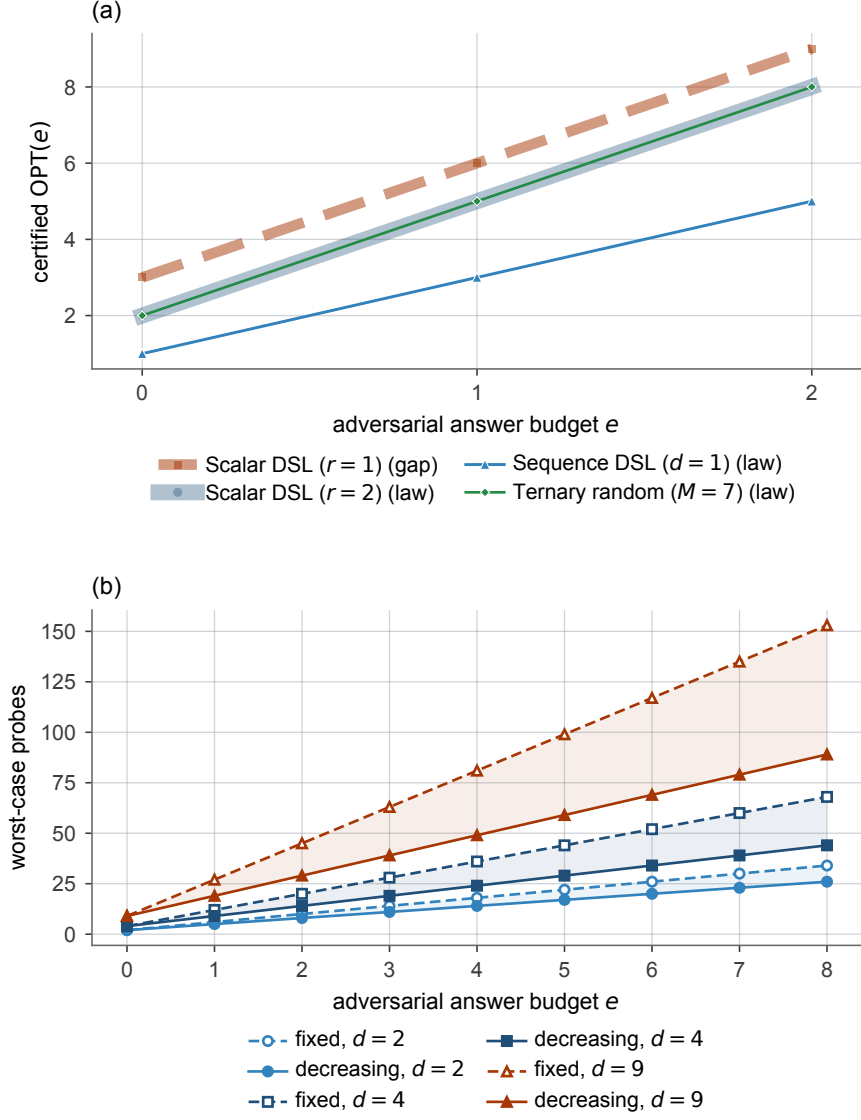


**Fig. 5** (a) Certified  $\text{OPT}(e)$  for four primary instances across  $e \in \{0, 1, 2\}$ . Solid lines mark matched laws under the original shallow rules; the dashed line marks a shallow-rule gap whose plotted points were established by expanded proofs. Isolation laws are reported separately in Section 11.2. The scalar radius-2 and random ternary tables are distinct instances satisfying the same law  $3e + 2$  and therefore coincide exactly; the former is drawn as a wide halo so that both remain visible. (b) The two repetition schedules as functions of the budget, at three basis depths, with the gap between them shaded. They differ by  $e(d - 1)$ , which is the entire advantage of spending certified lies

**Table 11** Historical shallow-rule scaling comparison at  $e = 10^6$ , evaluated symbolically. “Result” describes that rule’s bounds, and every row carries verified upper and lower evidence. The point and Boolean-function gaps shown here are closed by the isolation family theorem; Table 8 reports newly checked larger exact certificates.

| Certificate size and verification |     |     |              |       |             |            |
|-----------------------------------|-----|-----|--------------|-------|-------------|------------|
| Instance                          | $M$ | $Q$ | result       | nodes | payload (B) | verify (s) |
| Thresholds ( $n = 15$ )           | 16  | 15  | verified gap | 31    | 3,256       | 0.0006     |
| Thresholds ( $n = 63$ )           | 64  | 63  | verified gap | 127   | 10,586      | 0.0125     |
| Parity ( $k = 6$ )                | 64  | 63  | verified gap | 127   | 10,581      | 0.0105     |
| Point functions ( $n = 32$ )      | 32  | 32  | verified gap | 63    | 6,745       | 0.0036     |
| Boolean functions ( $b = 3$ )     | 256 | 8   | verified gap | 511   | 42,829      | 0.0117     |
| Sequence DSL ( $d = 2$ )          | 49  | 220 | exact law    | 62    | 10,385      | 0.0479     |

| Instance                      | Verified $L$ / $U$ at $e = 10^6$          |
|-------------------------------|-------------------------------------------|
| Thresholds ( $n = 15$ )       | $3 \cdot 10^6 + 2$ / $5 \cdot 10^6 + 4$   |
| Thresholds ( $n = 63$ )       | $3 \cdot 10^6 + 2$ / $7 \cdot 10^6 + 6$   |
| Parity ( $k = 6$ )            | $3 \cdot 10^6 + 2$ / $7 \cdot 10^6 + 6$   |
| Point functions ( $n = 32$ )  | $3 \cdot 10^6 + 2$ / $32 \cdot 10^6 + 31$ |
| Boolean functions ( $b = 3$ ) | $3 \cdot 10^6 + 2$ / $9 \cdot 10^6 + 8$   |
| Sequence DSL ( $d = 2$ )      | $3 \cdot 10^6 + 2$ (both)                 |

values: 69 have isolation laws, robustification is tight in 70, greedy is optimal in 71, and 91 have a positive adaptivity gap. Both parity-dimension-five limits remain unresolved.

A supplementary 13-cell  $e = 2$  suite, including two DSL tables, completes 12 expanded proofs; among them shallow laws close 6, robustification and greedy are each exact in 8, and 9 have a positive adaptivity gap. The depth-two sequence table reaches its specified 1800-second wall-clock cap before an expanded proof is produced. Its *portable* law nonetheless certifies  $\text{OPT}(2) = 8$ , illustrating the complementary roles of the two certificate families: the limit concerns expanded proof production, not the optimum.

## 11.7 Solver Ablations

Solver components were ablated on three representative cells.

Disabling the successor cache leaves the states visited, certificate size, and optimum unchanged. On two of three cells it has little effect on runtime (scalar DSL with  $r = 2$  at  $e = 2$ : 0.354 s to 0.357 s; the binary random table with  $(M, Q, s) = (8, 7, 23)$  at  $e = 1$ : 0.0077 s to 0.0082 s). For thresholds on [15] at  $e = 1$ , it increases solve time from 0.325 s to 2.078 s. The observed cache benefit is concentrated on states with high in-degree.

Supplying the certified shallow compact lower bound as the search floor  $\phi$  changes the visited state count in none of the three cells, and changes runtime by less than 2%. Algorithm 3 already computes pairwise and volume bounds at the root, so the compact floor ( $\phi = 5$  on two cells,  $\phi = 8$  on the third) adds nothing the iterative deepening had not already established.

## 11.8 Cross-Platform Agreement

A clean CPython 3.12.3, x86-64 Linux environment regenerates the primary results. After omitting timing measurements and environment metadata, every reported combinatorial quantity and all 30 tabular rows agree with the reference macOS results. The original shallow-law results for all 15 primary instances also agree. This replication covers the original primary suite; the isolation evaluation was run only on macOS.

## 12 Discussion

The results support trying bounded isolation-witness discovery before expanding the robust game. A successful witness certifies every budget, while failure for a particular noiseless basis leaves open the use of another basis or a fixed-budget certificate. Other residual potentials may cover the remaining parity and threshold cases. The multicover and protected-pair bounds are available as certified floors but are not evaluated at internal search states; integrating them could reduce search cost.

Certified optima also provide benchmarks for query-selection heuristics over finite concept classes (Section 3). Applying the method to diagnosis or program repair would require constructing a suitable catalog and validating the answer-error model. Formal verification of the theorem and executable checker remains a further direction.

## 13 Related Work

### *Search and identification with unreliable answers.*

The Rényi–Ulam liar game [1, 2] and its descendants [3–5, 10, 11] study a learner facing a bounded or stochastic number of false answers. Many classical formulations permit rich question families, including arbitrary subset queries, comparison queries, or general channels [30]. Our restriction to one explicit declared alphabet is not incidental: exact optimization over that alphabet is computationally hard in general (Section 9), which motivates independently checkable certificates. The volume argument in Theorem 29 follows the Berlekamp weight and volume tradition [31]; our version accommodates non-uniform branching and heterogeneous per-class budgets.

### *Fault-tolerant decision trees.*

Kenyon and Yao [6] introduced the bounded-fault decision-tree model used by Maharjan and Watson [7], who study a binary truth-table problem and prove the robustification inequality we reuse. Their Observation 3 also establishes  $D_e(\text{AND}_p) = p(e + 1) + e$ , a close precedent for the uniform-budget isolation mechanism. Restricted small-set liar search is directly relevant as well. Meng et al. [21, Theorem 3(1) and Lemma 10] prove that singleton queries require  $2M - 1$  questions with one lie. Fraknói et al. [22, Section 2, following Theorem 4] explicitly state  $RU_e^1(M) = (e + 1)M - 1$ , the point-family value used here for  $M \geq 2$ . Their broader results handle all subsets up to a specified size, and their Section 6 uses recursive computation to refute a conjecture about optimal queries. Our formulation differs in its finite many-valued restricted-query model with off-table answers, table-bound portable evidence, explicit obstruction on

failure, and measured comparison of synthesis with replay. Empirically, the inequality is not tight on approximately one quarter of the fixed sweep population.

The isolation rule transfers a singleton-style lower argument to a supplied subset of an arbitrary declared table. A checker must inspect every probe, including those unused by the upper strategy, before combining that subset with a noiseless tree. The residual formula in Theorem 17, its matching rule, and the explicit certificate interface describe this transfer.

### *Query-based learning.*

Exact learning from queries [23], malicious answers to membership queries [8], greedy active learning [32], generalized binary search and its noisy variants [9, 33], adaptive submodularity [34], and simultaneous learning and covering under adversarial noise [24] develop algorithms and performance guarantees under a range of structural, distributional, or noise assumptions. Agnostic and disagreement-based active learning [35, 36] likewise emphasize learning rates. These goals differ from certifying an exact finite minimax value under a hard global corruption budget. We compute that value for explicit tables and emit independently checkable upper and lower evidence; the greedy strategy is included as an empirical baseline rather than as a surrogate for this broader literature.

The star number of Hanneke and Yang [37] is witnessed by hypotheses that differ from a centre on one point each within a selected set of query points. Isolation instead requires the majority-block condition for every permitted probe. A star configuration on selected coordinates therefore does not by itself certify isolation on the full alphabet; Example 2 shows why added probes matter.

### *Decision-tree and covering optimisation.*

The zero-error binary special case is the minimum-height decision-tree problem for a fixed object-test matrix. Garey [12] studied this objective directly; Laber and Nogueira [13] established logarithmic inapproximability, and Bshouty and Makhoul [14] developed polynomial-time constructions and sharpened the approximation landscape using Set Cover hardness [29]. This work concerns the same worst-case-height objective. Proposition 36 supplies a self-contained equality reduction tailored to our many-valued table model. Related but distinct decision-tree work studies objectives such as total external path length or expected identification cost [38–40]; Moshkov [41] gives a broader treatment of decision-tree complexity. The non-adaptive side reduces to set multicover, whose greedy and LP theory is classical [27, 28]. We use that theory only to obtain a checkable dual, not an approximation guarantee.

Narodytska et al. [42] use Boolean satisfiability (SAT) to learn minimum-size decision trees on training data. That objective is distinct from the worst-case identification depth certified here, but it supplies a related AI setting in which exact tree optimisation is valuable.

### *Certificates and proof-carrying computation.*

Certifying algorithms [15], program checking [43, 44], and proof-carrying code [16, 17] established the discipline we follow. Contemporary combinatorial solvers instantiate

it: clausal proofs checked independently [18], and exact rational certificates for linear and integer programs [19, 20]. In machine learning, “certificate” more often denotes a robustness guarantee for a fixed model [45, 46], and “certification query complexity” denotes finding a small witness for a function value [47]. Our usage is the solver one: a replayable object establishing a statement about the query complexity of a game. The distinction is in the construction: the specific ingredients assembled here are an output-free table binding, all-budget laws paired with explicit obstructions, and adversary DAGs that answer every declared probe and close on analytic subset bounds.

Bogaerts et al. [48] provide independently checkable dominance and symmetry-breaking methods for combinatorial optimisation. The isolation certificate considered here provides a lower-witness rule specific to finite identification.

## 14 Limitations

The method requires an explicit finite behaviour table, constructed by evaluating every hypothesis on every declared probe. It assumes a deterministic target in the catalog and a valid global error budget. Guarantees apply to identification on the declared alphabet; they do not establish statistical generalization or transfer to a different probe set.

Isolation discovery can take exponential time. It leaves 6 of 15 primary and 32 of 101 sweep tables without a matching witness for the supplied basis. An exhausted search at one witness size does not exclude other bases or other exact laws. Verification is polynomial in the explicit table and supplied proof, but no general polynomial bound on optimality-certificate size is established.

The evaluation uses generated tables. The primary suite reaches 49 classes and 220 probes, the sweep reaches 32 classes and 31 probes, and the constructed certificates reach 256 classes. Both random evaluations use the same generator. Timings are descriptive observations under the protocols of Section 10; they do not establish performance on larger catalogs or other execution environments.

## 15 Conclusion

Portable and expanded certificates provide independently checkable optimality guarantees for finite identification under adversarial answer errors. Isolation witnesses extend the compact lower-bound rules to arbitrary witness size through an exact unequal-residual argument. On the 101-table sweep, compact coverage increases from 25 to 69 tables; combining portable and expanded evidence certifies 302 of 303 fixed-budget cells. The remaining challenge is to extend compact lower witnesses and control proof size on tables where isolation does not match the noiseless basis.

## Statements and Declarations

### Compliance with Ethical Standards

#### *Funding.*

This research did not receive funding.

### *Competing interests.*

Vikram Lex is the sole founder of KarLex AI, a C corporation with no external investors or revenue related to this work, and declares no other relevant financial or non-financial interests.

### *Ethics approval and consent.*

This theoretical and computational study involved no human participants, human samples, or animal experiments. Ethics approval, consent to participate, and participant consent for publication are not applicable.

## **Data and Code Availability**

### *Data availability.*

The generated behaviour-table specifications, fixed evaluation summaries, certificate artifacts, and reproduction instructions supporting the reported results are included in Online Resource 1. No personal or third-party datasets were used in the computational experiments.

### *Code availability.*

The Python source, independent certificate checkers, tests, and experiment runners are included in Online Resource 1. The certificate algorithms and tests use only the Python standard library. Figure generation has a separate plotting dependency documented in that archive.

### *Supplementary information.*

Online Resource 1 (ESM\_1.zip): Python implementation, independent certificate checkers, tests, fixed numerical evidence, 242 portable isolation certificates, and a worked exact-learning demonstration.

## **A Proofs**

Every statement below concerns a fixed finite behaviour table  $A : \mathcal{Q} \times \mathcal{C} \rightarrow \mathcal{Y}$  over its complete declared probe alphabet, with  $M = |\mathcal{C}|$  and  $Q = |\mathcal{Q}|$ , and a fixed integer budget  $e \geq 0$ . Recall from Section 2 that a probe may be repeated, that an answer matching no live class output is the catch-all outcome, and that we assume every pair of distinct classes is separated by some declared probe.

### **A.1 Sufficiency of the Mismatch Profile**

*Proof of Proposition 4* Fix a transcript  $\sigma = (q_1, y_1, \dots, q_t, y_t)$  and let  $S(\sigma)$  be its belief state (Definition 3). We claim the set of transcript extensions that are valid for a class  $c$  depends on  $\sigma$  only through  $\mu_\sigma(c)$ . Indeed, validity of the concatenation  $\sigma\tau$  for  $c$  is the single inequality

$$\mu_\sigma(c) + \mu_\tau(c) \leq e,$$

because the mismatch count is additive over transcript concatenation: it counts rounds, and the rounds of  $\sigma$  and of  $\tau$  are disjoint. Hence for any continuation strategy the family of targets

consistent with  $\sigma\tau$  is determined by  $S(\sigma)$  and  $\tau$  alone, and a continuation identifies the target from  $\sigma$  if and only if it does so from any  $\sigma'$  with  $S(\sigma') = S(\sigma)$ . Therefore the residual game value is a function of the belief state. Writing  $\text{OPT}(S)$  for that value gives (1). A state with at most one live class is already decided. Otherwise the learner may restrict to probes that refine the live set: for a non-refining probe the adversary can return the common live output and leave the state unchanged. The learner chooses among the refining probes, and the adversary chooses among their reachable answers. The answers that need to be considered are one per block of  $\pi_q$  meeting the live set, plus the catch-all when it leaves a class alive, since two answers inducing the same block induce the same successor by (2).

For (2) itself: a class  $c$  with  $\mu(c) = k$  moves to mismatch  $k$  if it agrees with the answer, i.e. if  $c \in \beta$ , and to  $k + 1$  otherwise, and is deleted if  $k + 1 > e$ . Collecting the classes that land at level  $k$  gives  $(S_k \cap \beta) \cup (S_{k-1} \setminus \beta)$  for  $k \geq 1$  and  $S_0 \cap \beta$  for  $k = 0$ .  $\square$

## A.2 Indexed Strategy Certificates

*Proof of Theorem 7* Let  $C$  be an accepted certificate with claimed height  $d$ , and let  $\text{nodes}(C)$  be its node set. Acceptance means, in the order the verifier applies them:

1. the binding carried by  $C$  equals the binding recomputed from the catalog (Definition 5), so  $C$  is about  $A$  and not about some other table;
2. node identifiers are distinct non-negative integers, the root identifier exists, and the root's class block is exactly  $(0, 1, \dots, M - 1)$ ;
3. every node's class block is a strictly increasing tuple of valid class indices;
4. every singleton block is a leaf, carrying no probe index and no children;
5. every non-singleton block carries a probe index  $j \in \{0, \dots, Q - 1\}$  such that  $\pi_{q_j}$  restricted to the block has at least two non-empty parts, i.e.  $q_j$  refines the block;
6. the declared child list has exactly as many entries as that restriction has parts, and the  $i$ th child's class block equals the  $i$ th part in canonical order (parts sorted by least member);
7. every child identifier resolves to a node in  $\text{nodes}(C)$ ;
8. the graph reachable from the root is acyclic, contains all of  $\text{nodes}(C)$ , and its recomputed height equals  $d$ .

Read  $C$  as a strategy: at a node with block  $L$  and probe  $q_j$ , ask  $q_j$ , observe the truthful output  $y = A(q_j, c^*)$ , and descend to the unique part of  $\pi_{q_j}|_L$  containing  $c^*$ . This is well defined by conditions 5 and 6, since the parts are precisely the fibres of  $y \mapsto \{c \in L : A(q_j, c) = y\}$  and they partition  $L$ . Since the root block contains  $c^*$ , the invariant  $c^* \in L$  is preserved along the walk. By condition 5 each step strictly reduces  $|L|$ , so the walk terminates at a leaf, and by condition 4 the leaf's block is  $\{c^*\}$ , which identifies the target. By condition 8 the walk has length at most  $d$ , and  $d$  is attained on some root-to-leaf path, so  $d$  is exactly the worst-case number of probes.

Note that concrete output values are needed only by the verifier, while it recomputes the canonical parts in condition 6; they do not appear in  $C$ . Two probes with the same restricted partition are interchangeable at that node, so the choice of representative index is immaterial to soundness.  $\square$

## A.3 Decreasing-Residual Robustification

*Proof of Theorem 9* Write  $F(h, r) = (h + 1)r + h$  with  $F(0, r) = 0$ , and let  $\Phi(h, r)$  denote the worst-case number of probes used by the navigator of Definition 8 started at a node of

remaining noiseless height  $h$  with residual budget  $r$ . We show  $\Phi(h, r) \leq F(h, r)$  by induction on  $h$ .

For  $h = 0$  the node is a leaf of the noiseless tree, hence a singleton class block, and the navigator halts:  $\Phi(0, r) = 0 = F(0, r)$ .

For  $h \geq 1$ , let  $Y$  be the set of labels of the node's outgoing edges and let  $a \in Y$  be the truthful outcome for the target, i.e. the label of the edge whose block contains  $c^*$ . The navigator repeats the node's probe until some label in  $Y$  has been observed  $r + 1$  times.

*Only  $a$  can reach the threshold.* Each observation of a label  $b \in Y \setminus \{a\}$  is a corrupted answer, since the truthful answer at this probe is  $a$ . Reaching the threshold with  $b$  would require  $r + 1$  corruptions while only  $r$  remain, a contradiction. Off-table answers are never entered in any label counter at all, so they cannot reach the threshold either. Hence the phase ends with  $a$  at count  $r + 1$ .

*Accounting.* Suppose the phase used  $r + 1 + \ell$  observations in total, so  $\ell$  of them were answers other than  $a$ . Each such answer is a corruption, so the target's residual budget when the navigator descends the  $a$ -edge is at least  $r - \ell$ ; the navigator sets it to exactly  $r - \ell$ , which is conservative, and  $\ell \leq r$ . The child has remaining noiseless height  $h - 1$ , so by induction

$$\Phi(h, r) \leq \max_{0 \leq \ell \leq r} [(r + 1 + \ell) + F(h - 1, r - \ell)].$$

Expanding, for each  $\ell$ ,

$$\begin{aligned} (r + 1 + \ell) + F(h - 1, r - \ell) &= r + 1 + \ell + h(r - \ell) + (h - 1) \\ &= (h + 1)r + h - (h - 1)\ell \\ &= F(h, r) - (h - 1)\ell \leq F(h, r), \end{aligned}$$

using  $h \geq 1$ . Taking  $h = d$  and  $r = e$  at the root gives  $\text{OPT}_A(e) \leq \Phi(d, e) \leq F(d, e)$ .

The argument uses only the predicate “the observed label equals the target's truthful label at this probe”, never the structure of  $\mathcal{Y}$ , so it holds for finite many-valued probes. The navigator stores the current node identifier, residual budget, phase observation count, and one counter per outgoing label of the current node. Its working state therefore depends on local branching and counter magnitudes, not only on  $d$ , but it never stores a robust belief state.  $\square$

*Remark 37* Two certificate types use this argument differently. The portable all-budget law verifier replays the basis strategy and lower witness, then compares affine expressions symbolically; its work is independent of any budget at which the accepted law is later evaluated. A separate fixed-budget threshold-strategy verifier checks  $(r + 1 + \ell) + F(h - 1, r - \ell) \leq F(h, r)$  for all  $1 \leq h \leq d$ ,  $0 \leq r \leq e$ , and  $0 \leq \ell \leq r$ , giving  $d(e + 1)(e + 2)/2$  comparisons.

## A.4 Pair and Blind-Triple Lower Laws

**Lemma 38** (Two residual classes) *Let  $c, c'$  be distinct classes separated by some declared probe, with residual budgets  $r, s \geq 0$ . The value of the game restricted to  $\{c, c'\}$  is exactly  $r + s + 1$ .*

*Proof* Let  $T(r, s)$  be that value. A probe on which  $c$  and  $c'$  agree induces a single block on the pair, so its truthful answer leaves the state unchanged; the adversary may always give it, hence such probes cannot help and  $T$  is determined by separating probes. For a separating probe the adversary has two useful replies: answering  $A(q, c)$  charges  $c'$ , answering  $A(q, c')$  charges  $c$ . (The catch-all charges both and is dominated, by Lemma 22.) Therefore

$$T(r, s) = 1 + \max\{T(r - 1, s), T(r, s - 1)\}, \quad T(-1, s) = T(r, -1) = 0,$$

where a negative residual means the corresponding class has been eliminated and the state is decided. Induction on  $r + s$  gives  $T(r, s) = r + s + 1$ : the base case  $r = s = 0$  gives  $T = 1 + \max\{0, 0\} = 1$ , and in general  $T(r, s) = 1 + \max\{r + s, r + s\} = r + s + 1$ . Repeating one separating probe  $r + s + 1$  times attains it, since after  $r + s + 1$  answers one of the two classes has accrued more than its residual budget.  $\square$

*Proof of Lemma 11* Take any separated pair and restrict the game to it. By Lemma 22 the restricted value is at most the full value. At the initial state both residual budgets equal  $e$ , so Lemma 38 gives  $2e + 1$ , whence  $\text{OPT}_A(e) \geq 2e + 1$ .  $\square$

**Lemma 39** (Blind triple with residual budgets) *Let  $\{a, b, c\}$  be pairwise separated by  $\mathcal{Q}$  with no declared probe inducing three distinct outputs on the triple, and let  $r_a, r_b, r_c \geq 0$  be residual budgets with  $S = r_a + r_b + r_c$ . The value of the game restricted to  $\{a, b, c\}$  is exactly  $S + 2$ .*

*Proof* Write  $V(r_a, r_b, r_c)$  for the value and induct on  $S$ .

Every probe restricted to the triple induces either one block (useless, as in Lemma 38) or exactly two, since three are excluded by hypothesis. A two-block partition isolates one class from the other two; by symmetry of the argument suppose it isolates  $c$ , so its outcomes charge  $\{c\}$  or  $\{a, b\}$ .

*Lower bound.* Consider the outcome charging only  $c$ , i.e. the adversary answers the value shared by  $a$  and  $b$ . If  $r_c \geq 1$  the child has all three classes live with residual sum  $S - 1$ , so by induction it is worth  $(S - 1) + 2 = S + 1$ . If  $r_c = 0$  then  $c$  is eliminated and the child is a separated pair with residual budgets  $r_a, r_b$ , worth  $r_a + r_b + 1 = S + 1$  by Lemma 38. Either way the adversary has a reply worth  $S + 1$ , and this holds for *every* probe. Hence  $V \geq 1 + (S + 1) = S + 2$ .

*Upper bound.* Fix any probe refining the triple, isolating  $c$  say. Its shared-output child is worth exactly  $S + 1$  by the computation above. Its isolated-output child charges both  $a$  and  $b$ . If both survive, the child has residual sum  $S - 2$  and is worth  $S$  by induction; if exactly one survives, the surviving pair is worth at most  $(S - 1) + 1 = S$  by Lemma 38; if neither survives, the child is decided and worth 0. So the worst child of this probe is worth  $S + 1$  and  $V \leq 1 + (S + 1) = S + 2$ .

The base case  $S = 0$  is covered by the same computation: both children are decided or are pairs with zero residuals, giving  $V = 2$ .  $\square$

*Proof of Theorem 13* Restrict to the blind triple, set every residual budget to  $e$  in Lemma 39, and apply Lemma 22. The restricted value  $3e + 2$  lower-bounds  $\text{OPT}_A(e)$ .  $\square$

*Proof of Corollary 14* (1) A single class needs no probe, and no lower bound exceeds 0. (2) A probe separating every pair is a depth-one strategy, so Theorem 9 gives  $\text{OPT}_A(e) \leq F(1, e) = 2e + 1$ , while Lemma 11 gives the matching lower bound; note a depth-one strategy exists only if  $M \geq 2$ , so a separated pair exists. (3) A depth-two strategy gives  $\text{OPT}_A(e) \leq F(2, e) = 3e + 2$  and Theorem 13 gives the matching lower bound.  $\square$

*Proof of Proposition 15* Algorithm 1 performs three searches, each exhaustive over the explicit table.

*Depth one.* It tests every probe for whether  $\pi_q$  has  $M$  blocks. A depth-one strategy exists if and only if some probe does, so this test is exact, at cost  $O(MQ)$ .

*Depth at most two.* It tests every probe  $q$  as a root and, for each non-singleton block  $\beta$  of  $\pi_q$ , searches all probes for one whose restriction to  $\beta$  has  $|\beta|$  blocks. A strategy of depth at most two exists if and only if some root admits such a completion for every non-singleton block, since a depth-two strategy is exactly a root probe together with one further probe per block that must fully separate it. The search is therefore exact, at cost  $O(MQ^2)$ .

*Blind triple.* It enumerates all  $\binom{M}{3}$  triples and, for each, scans every probe to test pairwise separation and the absence of a three-way split, at cost  $O(M^3Q)$ . This is exact.

Consequently the algorithm reports an exact law precisely when one of the three regimes of Corollary 14 holds, which is the completeness claim, and it makes no claim about deeper matched laws. On the gap branch the upper bound comes from a deterministic greedy noiseless tree, verifiable by Theorem 7 and Theorem 9, and the lower bound is  $L(e) = \max\{\lambda_0, \text{law}(w)(e)\}$  where  $w$  is the strongest witness found and  $\lambda_0$  is the noiseless floor. Both terms are sound: the witness law by Lemma 11 or Theorem 13, and the floor because the exhaustive depth-one and depth-two searches failing certifies  $\text{OPT}_A(0) \geq 3$  (respectively  $\geq 2$  when only the depth-one search failed). Finally, the obstruction category is determined by the two Boolean search outcomes. The independent verifier recomputes both searches and rejects a category that disagrees with them, together with a certificate whose recorded values of “a depth-one strategy exists”, “a depth-two strategy exists”, and “a blind triple exists” do not match the recomputation.  $\square$

## A.5 Structural Lemmas for the Expanded Solver

*Proof of Lemma 21* Let  $q$  refine  $L(S)$ , so  $\pi_q|_L$  has at least two non-empty blocks, and let  $\beta$  be the block consistent with the answer (with  $\beta = \emptyset$  for the catch-all). Every class in  $L \setminus \beta$  has its mismatch count incremented by (2), and  $L \setminus \beta \neq \emptyset$  because  $q$  refines. So either some class exceeds  $e$  and leaves the live set, strictly decreasing  $|L|$ , or no class leaves and  $\sum_{c \in L} \mu(c)$  strictly increases while  $|L|$  is unchanged. In the first case the first coordinate of  $(|L|, e|L| - \sum_{c \in L} \mu(c))$  decreases; in the second the first coordinate is fixed and the second strictly decreases. Both coordinates are non-negative integers bounded by  $M$  and  $eM$  respectively, so the lexicographic order is well founded and no state can recur along a path: the reachable graph over refining probes is a finite DAG, and (1) is a well-defined recursion needing no depth parameter.

For the last claim, if  $q$  does not refine  $L$  then all live classes share one output on  $q$ ; the adversary can answer it, leaving  $S$  unchanged, so the probe consumes a round without changing the state and is strictly wasteful in the worst case. Excluding such probes therefore does not change  $\text{OPT}$ .  $\square$

*Proof of Lemma 22* Suppose  $S \preceq S'$  and let  $T$  be a strategy solving  $S'$  within  $k$  probes. Run  $T$  from  $S$ . Fix any answer sequence  $\tau$  realisable from both states and let  $\Delta_\tau(c)$  be the number of answers in  $\tau$  that  $c$  disagrees with; this depends only on  $\tau$  and the table, not on the starting state. A class  $c$  is live at the end of  $\tau$  from  $S$  iff  $c \in L(S)$  and  $\mu_S(c) + \Delta_\tau(c) \leq e$ , and similarly from  $S'$ . Since  $L(S) \subseteq L(S')$  and  $\mu_S(c) \geq \mu_{S'}(c)$  on  $L(S)$ , every class live from  $S$  after  $\tau$  is also live from  $S'$  after  $\tau$ . As  $T$  leaves at most one class live from  $S'$  at each of its leaves, it leaves at most one live from  $S$  there too, so  $T$  solves  $S$  within  $k$  probes. Hence  $\text{OPT}(S) \leq \text{OPT}(S')$ .  $\square$

*Proof of Corollary 23* Fix a state  $S$  and a probe  $q$ , let  $\beta \neq \emptyset$  be any block of  $\pi_q$  meeting  $L(S)$ , and compare the successors  $S_\beta$  (a genuine answer) and  $S_\emptyset$  (the catch-all). By (2), every class in  $\beta$  keeps its mismatch count in  $S_\beta$  but is incremented in  $S_\emptyset$ , while classes outside  $\beta$  are incremented in both. Hence  $L(S_\emptyset) \subseteq L(S_\beta)$  and  $\mu_{S_\emptyset}(c) \geq \mu_{S_\beta}(c)$  for every  $c \in L(S_\emptyset)$ , i.e.  $S_\emptyset \preceq S_\beta$ . By Lemma 22,  $\text{OPT}(S_\emptyset) \leq \text{OPT}(S_\beta)$ , so an adversary maximising OPT never strictly prefers the catch-all and may always be taken to answer a genuine class output.  $\square$

*Proof of Corollary 24* By (2) the successor state depends on the answer only through the block  $\beta \cap L(S)$ , and the set of blocks available at  $S$  is exactly the set of parts of  $\pi_q|_{L(S)}$ . Hence two probes with equal  $\pi_q|_{L(S)}$  induce identical successor sets at  $S$ , and two probes with equal  $\pi_q$  induce identical successor sets at every state. Collapsing probes by (restricted) partition therefore preserves the value of (1) exactly. For certificates the collapse must be undone: an adversary certificate is required to answer every *declared* probe, so each collapsed representative's reply is expanded to all probes sharing its partition, which is legitimate because they induce the same successor.  $\square$

## A.6 Volume Bounds

*Proof of Theorem 29* Suppose  $S$  is solvable within  $d$  probes and let  $T$  be a solving strategy of depth at most  $d$ . Pad each shorter root-to-leaf path with virtual unary no-op edges until every path has length  $d$ . These virtual edges leave the survivor set unchanged and require no declared probe. Perform a random walk from the root, choosing uniformly among the  $b_t \leq B$  outgoing edges at each real node and taking the sole edge at each virtual node. Leaf probabilities sum to 1.

Fix a live class  $c$  with residual budget  $r(c)$ . At a real node where  $c$  remains live, exactly one of the  $b_t$  edges carries  $c$ 's output, so the walk disagrees with probability  $(b_t - 1)/b_t \leq (B - 1)/B$ . A virtual step has disagreement probability zero. Hence the number of disagreements accumulated by  $c$  over the  $d$  positions is stochastically dominated by  $\text{Bin}(d, (B - 1)/B)$ , and

$$\Pr[c \text{ survives}] \geq \Pr[\text{Bin}(d, (B - 1)/B) \leq r(c)] = \frac{1}{B^d} \sum_{j=0}^{\min(r(c), d)} \binom{d}{j} (B - 1)^j = \frac{V_d^B(r(c))}{B^d}.$$

Because  $T$  solves  $S$ , each leaf leaves at most one class live, so the events  $\{c \text{ survives}\}$  for distinct  $c \in L(S)$  are pairwise disjoint. Summing,

$$\frac{1}{B^d} \sum_{c \in L(S)} V_d^B(r(c)) \leq 1,$$

which is  $W_B(S, d) \leq B^d$ . Non-uniform branching only lowers each disagreement probability, so the conclusion holds with  $B$  any upper bound on the fan-out of every probe at every reachable descendant state. The widest restricted fan-out over the live set, plus one when the catch-all can still leave a class alive, is such a bound because live sets only shrink and mismatch counts only grow.

The least  $d$  with  $W_B(S, d) \leq B^d$  is therefore a valid lower bound. The predicate is monotone in  $d$ : by the Pascal identity  $\binom{d+1}{j} = \binom{d}{j} + \binom{d}{j-1}$ ,

$$V_{d+1}^B(r) \leq B V_d^B(r) \quad \text{hence} \quad W_B(S, d+1) \leq B W_B(S, d),$$

so once  $W \leq B^d$  holds it continues to hold for all larger  $d$ , and every  $d$  below the threshold is refuted.  $\square$

*Proof of Corollary 30* Let  $J \subseteq L(S)$  with  $|J| \geq 2$  and let  $S|_J$  be the state with live set  $J$  and the same mismatch counts. Then  $S|_J \preceq S$ , so  $\text{OPT}(S|_J) \leq \text{OPT}(S)$  by Lemma 22, and Theorem 29 applied to  $S|_J$  with  $B_J$  the fan-out bound on  $J$  gives a lower bound on  $\text{OPT}(S|_J)$ , hence on  $\text{OPT}(S)$ . Since  $B_J$  can be strictly smaller than  $B$  and appears in the exponent, the resulting bound can exceed the one obtained from the full live set.  $\square$

## A.7 Covering Bounds

*Proof of Theorem 31* Fix a live reference  $i$  and let  $T$  be a strategy solving  $S$  in  $d$  probes. Run  $T$  against the oracle that answers  $A(q, i)$  at every probe  $q$  it asks. This is a legal oracle for target  $i$ : it never disagrees with  $i$ , so  $\mu(i)$  stays at its initial value and  $i$  remains live throughout, and in particular the transcript is valid for  $i$  with zero corruptions. Since  $T$  solves  $S$ , the run reaches a leaf at which at most one class is live; as  $i$  is live, the leaf identifies  $i$ , and every other class  $j \in L(S) \setminus \{i\}$  is dead there.

For  $j$  to be dead it must have accumulated  $e - \mu(j) + 1 = d_j$  fresh disagreements along the run. A fresh disagreement for  $j$  occurs at a probe  $q$  only if  $j$ 's output differs from the answer given, i.e. only if  $A(q, j) \neq A(q, i)$ : only probes separating  $j$  from  $i$  can charge  $j$ . Therefore the multiset  $P$  of probes asked along this single root-to-leaf path satisfies

$$|\{q \in P : A(q, j) \neq A(q, i)\}| \geq d_j \quad \text{for every } j \in L(S) \setminus \{i\},$$

i.e.  $P$  is a feasible multiset for the multicover instance with incidence  $a_{q,j} = \mathbb{1}\{A(q, j) \neq A(q, i)\}$  and demands  $d_j$ . As  $|P| \leq d$ , we get  $d \geq \text{MC}_i(S)$ , and maximising over  $i$  gives the claim.

For the dual,  $\text{MC}_i$  is the optimum of  $\min\{\mathbf{1}^\top x : Ax \geq d, x \in \mathbb{Z}_{\geq 0}^Q\}$ , which is at least the LP optimum, whose dual is  $\max\{d^\top y : A^\top y \leq \mathbf{1}, y \geq 0\}$ . Any feasible  $y$  therefore gives  $\text{MC}_i \geq \lceil d^\top y \rceil$ , the ceiling being valid because  $\text{MC}_i$  is an integer. A verifier need only recompute  $A$  from the catalog, confirm  $y \geq 0$ , check the  $Q$  inequalities  $\sum_j a_{q,j} y_j \leq 1$  in exact rational arithmetic, and compare  $\lceil d^\top y \rceil$  against the claimed depth.  $\square$

*Proof of Corollary 32* If no probe separates more than  $t$  members of  $J$  from  $i$  then for every  $q$ ,  $\sum_{j \in J} a_{q,j} \leq t$ , so  $y = \mathbf{1}_J/t$  satisfies  $A^\top y \leq \mathbf{1}$  and is feasible. Its objective is  $\sum_{j \in J} d_j/t$ , giving  $\text{OPT} \geq \lceil (\sum_{j \in J} d_j)/t \rceil$ . At the initial state all  $\mu(j) = 0$ , so  $d_j = e + 1$  and the bound is  $\lceil (e + 1)|J|/t \rceil$ .  $\square$

*Proof of Theorem 33* Fix a state, a live reference  $i$  with  $\Lambda = e - \mu(i)$ , and a live protégé  $j$ . Let  $T$  solve the state in  $d$  probes and run it against the described oracle. Call a step *lying* if the probe separates  $i$  from  $j$  and fewer than  $\Lambda$  lying steps have occurred, in which case the answer is  $A(q, j)$ ; otherwise the step is *truthful* with answer  $A(q, i)$ .

Each lying step disagrees with  $i$  exactly once, so  $\mu(i)$  grows by one per lying step and by nothing on truthful steps. Since at most  $\Lambda$  lying steps occur,  $\mu(i) \leq e$  throughout and  $i$  stays live; so the run reaches a leaf identifying  $i$ , and every other live class is dead there.

Let  $t_q$  and  $l_q$  count truthful and lying steps at probe  $q$ , so  $d = \sum_q (t_q + l_q)$ . Three families of constraints hold.

- (1)  $j$  can be charged only at a step whose answer differs from  $A(q, j)$ . A lying step answers  $A(q, j)$  and so never charges  $j$ ; a truthful step charges  $j$  exactly when  $A(q, i) \neq A(q, j)$ . Since  $j$  must be charged  $d_j$  times,  $\sum_{q: A(q, i) \neq A(q, j)} t_q \geq d_j$ .

- (2) Any other live  $k$  is charged by a truthful step at  $q$  iff  $A(q, k) \neq A(q, i)$  and by a lying step at  $q$  iff  $A(q, k) \neq A(q, j)$ , so  $\sum_q \mathbb{1}\{A(q, k) \neq A(q, i)\} t_q + \sum_q \mathbb{1}\{A(q, k) \neq A(q, j)\} l_q \geq d_k$ .
- (3) The run must contain  $\Lambda$  lying steps: if fewer occur, the oracle was truthful throughout the remaining steps, so no answer ever disagreed with  $j$  at a separating probe and  $j$  was never charged there; combined with (1) this would leave  $j$  live at the leaf alongside  $i$ , contradicting that  $T$  solves the state. Hence  $\sum_{q: A(q, i) \neq A(q, j)} l_q \geq \Lambda$ .

Thus  $(t, l)$  is a feasible point of the covering program with those three constraint families and objective  $\sum_q (t_q + l_q)$ , so  $d$  is at least that program's optimum, which is therefore a lower bound on OPT. Its LP dual has one variable per constraint,  $y_0$  for (1),  $y_k$  for each (2), and  $y_\Lambda$  for (3), with one 0/1 constraint per  $t_q$  column and one per  $l_q$  column, and objective  $d_j y_0 + \sum_k d_k y_k + \Lambda y_\Lambda$ ; any feasible dual point certifies the bound and is checked against every declared probe in exact rational arithmetic.  $\square$

## A.8 Adversary Certificates

*Proof of Proposition 27* We prove by induction on the remaining depth  $m$  that every verified node with belief state  $S$  and remaining depth  $m$  satisfies  $\text{OPT}(S) > m$ .

*Closed nodes.* Verification of a pairwise closure naming live classes  $c, c'$  recomputes their mismatch counts from the node's state and checks  $(e - \mu(c)) + (e - \mu(c')) + 1 > m$ ; by Lemma 38 the left side is OPT of the restricted pair state, and by Lemma 22 that lower-bounds  $\text{OPT}(S)$ , so  $\text{OPT}(S) > m$ . Verification of a volume closure naming a subset  $J$  and a branching  $B$  recomputes the subset's fan-out, rejects a claimed  $B$  below it, and checks  $W_B(S|_J, m) > B^m$ ; by Theorem 29 the subset state is not solvable in  $m$  probes, and by Lemma 22 neither is  $S$ .

*Open nodes.* The verifier requires  $m \geq 1$  (a node at  $m = 0$  without a closure is rejected), requires the state to keep at least two classes live, and requires exactly one reply per declared probe index with no duplicates. For each reply it recomputes the successor of the node's state under that concrete answer directly from the catalog and requires it to equal the declared child's state, and requires the child's remaining depth to be  $m - 1$ . Now let  $T$  be any strategy and suppose it is at this node. If  $T$  halts, at least two classes are consistent with the transcript, so it has not identified the target. Otherwise  $T$  asks some declared probe  $q$ ; the certificate supplies a reply for  $q$ , which is a genuine output of some class (Corollary 23 explains why this is possible) and hence a legal oracle answer, and the resulting state is the child's, with remaining depth  $m - 1$ . By induction  $\text{OPT}(\text{child}) > m - 1$ , so  $T$  cannot finish within  $m - 1$  further probes, and  $\text{OPT}(S) > m$ . Well-foundedness is immediate since the remaining depth strictly decreases along every edge, and the verifier additionally rejects certificates carrying nodes unreachable from the root.

*Root.* The verifier checks that the root state is either  $S^{\text{init}}$  or, when a class subset is declared, that subset with every member at mismatch zero. In the latter case Lemma 22 transfers the bound to the full game. The root's remaining depth must also equal the claimed refuted depth  $k$ . Hence  $\text{OPT}_A(e) > k$ .

If a strategy certificate of depth  $k + 1$  also verifies, then  $\text{OPT}_A(e) = k + 1$ . Following the stored reply for each probe chosen by the strategy produces an explicit legal transcript ending with at least two classes live. This transcript is the executable form of the lower bound.  $\square$

## A.9 Non-Adaptive Codes

*Proof of Theorem 34* A probe multiset  $P = (p_1, \dots, p_n)$  assigns to each class  $c$  the codeword  $x_c = (A(p_1, c), \dots, A(p_n, c))$ , and a transcript is a word  $y \in \mathcal{Y}^n$  valid for  $c$  iff  $\text{dist}(y, x_c) \leq e$ .

*Sufficiency.* If  $\text{dist}(x_c, x_{c'}) \geq 2e + 1$  for all  $c \neq c'$  then the radius- $e$  balls around codewords are pairwise disjoint, so every transcript is within  $e$  of at most one codeword and nearest-codeword decoding recovers the target.

*Necessity.* Suppose  $\text{dist}(x_c, x_{c'}) = \delta \leq 2e$  for some  $c \neq c'$ . Let  $D$  be the set of coordinates where they differ, choose  $F \subseteq D$  with  $|F| = \min(e, \delta)$ , and set  $y_p = x_{c'}[p]$  for  $p \in F$  and  $y_p = x_c[p]$  otherwise. Then  $\text{dist}(y, x_c) = |F| = \min(e, \delta) \leq e$  and  $\text{dist}(y, x_{c'}) = \delta - |F| = \max(0, \delta - e) \leq e$  since  $\delta \leq 2e$ . So  $y$  is valid for both classes and no decoder can separate them.  $\square$

The necessity construction is also a concrete confusion witness. Its checker rebuilds the code from the catalog and verifies that both distances equal their declared values and are at most  $e$ .

*Proof of Proposition 35* By Theorem 34 the non-adaptive optimum is the least  $n$  for which some multiset of size  $n$  separates every class pair at least  $2e + 1$  times, i.e. the integer program  $\min\{\mathbf{1}^\top x : Ax \geq (2e+1)\mathbf{1}, x \in \mathbb{Z}_{\geq 0}^Q\}$  with  $a_{q,p} = \mathbb{1}\{q \text{ separates pair } p\}$ . Its LP relaxation has dual  $\max\{(2e+1)\mathbf{1}^\top y : A^\top y \leq \mathbf{1}, y \geq 0\}$ , so any feasible  $y$  gives  $\text{OPT}_A^{\text{na}}(e) \geq (2e+1) \sum_p y_p$ , and the ceiling is valid by integrality of  $\text{OPT}_A^{\text{na}}(e)$ . Verification recomputes each probe's separation mask from the catalog, checks  $y \geq 0$  and the  $Q$  load inequalities in exact rationals, and requires the claimed bound to equal  $\lceil (2e+1) \sum_p y_p \rceil$ . Equality rather than inequality prevents a certificate from understating its own rounding.  $\square$

## A.10 Hardness

*Proof of Proposition 36* Let  $(U, \mathcal{S})$  be a coverable Set Cover instance with  $U = \{0, \dots, n-1\}$  and  $\mathcal{S} = (S_1, \dots, S_m)$ . Build a catalog of  $n+1$  hypotheses over the probe set  $\{1, \dots, m\}$  by explicit output rows:

$$A(q_k, c_0) = 0, \quad A(q_k, c_u) = \begin{cases} 1 + u, & u \in S_k, \\ 0, & u \notin S_k, \end{cases}$$

for  $u \in U$ . The construction is  $O(nm)$  and every row is distinct because the instance is coverable, so the  $n+1$  hypotheses form  $n+1$  semantic classes. Take  $e = 0$ .

$\text{OPT}_A(0) \geq \tau$ , the minimum cover size. Let  $T$  be a noiseless strategy of depth  $t$  and follow the transcript in which every answer is 0; this is the truthful transcript for the anchor  $c_0$ , so  $T$  must identify  $c_0$  at its end. Let  $K$  be the set of probes asked along it. If some  $u \in U$  is not covered by  $\{S_k : k \in K\}$  then  $A(q_k, c_u) = 0$  for all  $k \in K$ , so  $c_u$  has the same transcript as  $c_0$  and  $T$  cannot have identified  $c_0$ . Hence  $\{S_k : k \in K\}$  covers  $U$  and  $t \geq |K| \geq \tau$ .

$\text{OPT}_A(0) \leq \tau$ . Ask the probes of a minimum cover, non-adaptively. The anchor produces the all-zero word. Each  $c_u$  produces a word containing the symbol  $1 + u$  at some coordinate, since the cover contains a set holding  $u$ ; these symbols are distinct across  $u$  and distinct from 0, so all  $n+1$  words are pairwise distinct and the target is identified. Thus  $\text{OPT}_A(0) = \tau$ , and since minimum Set Cover is NP-hard [49], so is computing  $\text{OPT}_A(e)$  exactly at  $e = 0$ ; a polynomial-time exact algorithm would give  $\text{P} = \text{NP}$ .  $\square$

*Remark 40* (Scope of the equality reduction) The map is an *equality* reduction, not a gap-preserving one, so it does not itself transfer a Set Cover approximation threshold to  $\text{OPT}_A(e)$ . Logarithmic inapproximability for the binary minimum-height decision-tree problem is independently known [13, 14]; the limitation here concerns this particular proof, not the prior-art landscape. Separately, Proposition 36 says nothing about certificate size: our verification is

polynomial in the explicit table and the supplied proof, and no polynomial bound on proof size in a succinct description of the game is claimed or implied.

## A.11 Verification Cost Accounting

All costs below exclude shared catalog and table reconstruction. Recomputing outputs of the  $N$  catalog hypotheses on  $Q$  probes requires  $O(NQ)$  hypothesis evaluations, with the cost of each evaluation depending on the hypothesis. The current semantic validator also checks agreement within classes and separation between representatives, requiring  $O(NQ + M^2Q)$  table-entry operations. These costs distinguish the raw catalog size  $N$  from the semantic-class count  $M$ .

- *Indexed strategy* (Theorem 7): each node is visited once and its restricted partition recomputed in  $O(|L|)$ , so  $O(MQ_{\text{used}})$  overall with  $Q_{\text{used}} \leq Q$ ; height and reachability are a single DAG traversal.
- *Portable law*: the basis and lower-witness checks above, followed by a symbolic affine comparison; the cost is independent of the budget at which the law is evaluated.
- *Fixed-budget threshold strategy* (Theorem 9): the basis replay plus  $d(e+1)(e+2)/2$  integer comparisons.
- *Pair witness*:  $O(Q)$ . *Blind triple*:  $O(Q)$  for a named triple; the exhaustive search that *finds* one is  $O(M^3Q)$  and is a producer cost.
- *Isolation witness*:  $O(kQ)$  for a supplied  $k$ -class witness, in addition to the shared semantic-table checks and indexed-tree replay. Subset discovery is a producer cost and can be exponential; its resource limit is not an exhaustive-absence certificate.
- *Volume certificate*:  $O(MQ)$  to recompute the subset fan-out plus  $O(d)$  big-integer operations for the character.
- *Multicover and protected-pair certificates*:  $O(MQ)$  exact-rational operations, being one inequality per declared probe (two per probe for  $P^+$ ).
- *Adversary certificate* (Proposition 27): each node requires one recomputed transition per declared probe, each costing  $O(M)$  with a bitmask state, hence  $O(\text{nodes} \cdot Q \cdot M)$ ; closures add  $O(M+m)$  each.
- *Non-adaptive code*:  $O(nM)$  to rebuild the code and  $O(M^2n)$  for pair distances, plus  $O(MQ)$  rational operations for the dual.

These are polynomial in the explicit table and the proof, which is the claim of Section 4, and they do not bound proof size.

## B Instance Catalog

### B.1 Program-Derived Tables

#### *Scalar DSL.*

A program is a source in  $(x, y, \text{zero})$  followed by a sequence of operations applied left-to-right to an accumulator, with both inputs readable at every step. The nine operations, in canonical order, are **abs**, **add\_x**, **add\_y**, **double**, **max\_y**, **min\_x**, **negate**, **sub\_x**, **sub\_y**. The depth-one catalog is therefore  $3 \times (1+9) = 30$  programs. A radius- $r$  probe set is the full integer grid  $\{-r, \dots, r\}^2$  in row-major order, so  $|\mathcal{Q}| = (2r+1)^2$ :

nine probes at  $r = 1$  and twenty-five at  $r = 2$ . Behavioural reduction collapses the 30 programs to 16 semantic classes in both cases; the two tables differ only in their alphabet, which is what makes the radius control of Section 11.1 a clean comparison.

### *Sequence DSL.*

A program is a pipeline of primitives applied to an integer tuple. The eight primitives, in canonical order, are **add1**, **even**, **negate**, **positive**, **reverse**, **rotate**, **sort**, **unique**. Depth one gives  $1 + 8 = 9$  programs and depth two gives 73. The probe set is all tuples over  $\{-2, \dots, 2\}$  of length  $0, \dots, 3$  (156 of them) together with 64 deterministically sampled length-4 tuples drawn from a seeded generator, for  $|\mathcal{Q}| = 220$ . Depth one yields 9 semantic classes and depth two yields 49.

## B.2 Classical Concept Tables

Each of the following is built from an explicit output matrix; behavioural reduction is then a no-op, so  $N = M$ .

- *Thresholds on  $[n]$* : classes are the  $n + 1$  concepts  $x \geq t$  for  $t = 0, \dots, n$ ; probes are the  $n$  domain points.
- *Parities of dimension  $k$* : classes are the  $2^k$  hidden vectors; probes are the  $2^k - 1$  non-zero query vectors, answering  $\langle \text{hidden}, \text{probe} \rangle \bmod 2$ .
- *Point functions on  $[n]$* : classes are the  $n$  one-hot indicators; probes are the  $n$  domain points.
- *All Boolean functions on  $b$  bits*: classes are all  $2^{2^b}$  truth tables; probes are the  $2^b$  inputs.
- *Monotone conjunctions on  $k$  variables*: classes are the  $2^k$  variable subsets including the empty conjunction; probes are the  $2^k$  assignments.
- *Random tables*: given a shape (class count, probe count, arity) and a seed, rows are drawn uniformly without replacement from  $\{0, \dots, \text{arity} - 1\}^{\text{probes}}$  until the declared number of distinct rows is obtained.

The extended sweep instantiates threshold domains  $3, 5, \dots, 19$ ; parity dimensions  $2, \dots, 5$ ; point counts  $3, \dots, 10$ ; monotone conjunctions on  $2, 3, 4$  variables; all Boolean functions on 1 and 2 bits; and the three random shapes  $(6, 5, 2)$ ,  $(8, 7, 2)$ ,  $(7, 6, 3)$  crossed with seeds  $0, \dots, 24$ , for  $26 + 75 = 101$  instances.

## C Complete Result Tables

Table 12 gives every primary cell. Two patterns are visible in it that the main text discusses only in aggregate. First, at  $e = 0$  the shallow compact lower bound  $L$  is the exhaustive noiseless floor  $\lambda_0 = 3$  on every unresolved instance, including the 10-class point-function table, where the optimum is 9, confirming that at zero budget it is the floor and not the affine witness that binds. Second,  $R_{\text{fix}}$  and  $R_{\text{dec}}$  coincide at  $e = 0$  by definition and diverge by exactly  $d - 1$  at  $e = 1$ , which is the schedule difference reported in the main text.

Table 13 records the supplementary evaluation at error budget two.

**Table 12** The original primary matrix with shallow-rule bounds, all 30 cells.  $d$  is the verified noiseless basis depth;  $L$  and  $U$  are the shallow compact lower and upper bounds;  $R_{\text{fix}} = (2e + 1)d$  and  $R_{\text{dec}} = F(d, e)$  are the two repetition schedules;  $G$  is greedy depth; OPT is the certified adaptive optimum; “states” is the exact search’s visited state count; and  $\text{OPT}^{\text{na}}$  is the certified non-adaptive optimum. The shallow column records either a matched law or the verified obstruction that no depth- $\leq 2$  strategy exists. For random tables, each triple gives  $(M, Q, \text{seed})$ . Rows marked \* carry a matched all-budget law, in which case  $L = U = \text{OPT}$ . Isolation certificates extend coverage from 4 to 9 primary tables (Section 11.2); this table retains the original method’s bounds.

| Instance                          | shallow     | $e$ | $d$ | $L$ | $U$ | $R_{\text{fix}}$ | $R_{\text{dec}}$ | $G$ | OPT | states | $\text{OPT}^{\text{na}}$ |
|-----------------------------------|-------------|-----|-----|-----|-----|------------------|------------------|-----|-----|--------|--------------------------|
| Scalar DSL ( $r = 1$ )            | depth $> 2$ | 0   | 3   | 3   | 3   | 3                | 3                | 3   | 3   | 19     | 3                        |
| Scalar DSL ( $r = 1$ )            | depth $> 2$ | 1   | 3   | 5   | 7   | 9                | 7                | 7   | 6   | 782    | 8                        |
| Scalar DSL ( $r = 2$ )*           | law         | 0   | 2   | 2   | 2   | 2                | 2                | 2   | 2   | 6      | 2                        |
| Scalar DSL ( $r = 2$ )*           | law         | 1   | 2   | 5   | 5   | 6                | 5                | 5   | 5   | 372    | 6                        |
| Sequence DSL ( $d = 1$ )*         | law         | 0   | 1   | 1   | 1   | 1                | 1                | 1   | 1   | 1      | 1                        |
| Sequence DSL ( $d = 1$ )*         | law         | 1   | 1   | 3   | 3   | 3                | 3                | 3   | 3   | 47     | 3                        |
| Sequence DSL ( $d = 2$ )*         | law         | 0   | 2   | 2   | 2   | 2                | 2                | 2   | 2   | 9      | 3                        |
| Sequence DSL ( $d = 2$ )*         | law         | 1   | 2   | 5   | 5   | 6                | 5                | 5   | 5   | 13,898 | 6                        |
| Binary random (6, 5, 11)          | depth $> 2$ | 0   | 3   | 3   | 3   | 3                | 3                | 3   | 3   | 5      | 3                        |
| Binary random (6, 5, 11)          | depth $> 2$ | 1   | 3   | 5   | 7   | 9                | 7                | 7   | 7   | 174    | 9                        |
| Binary random (8, 7, 23)          | depth $> 2$ | 0   | 3   | 3   | 3   | 3                | 3                | 3   | 3   | 7      | 4                        |
| Binary random (8, 7, 23)          | depth $> 2$ | 1   | 3   | 5   | 7   | 9                | 7                | 7   | 6   | 291    | 10                       |
| Ternary random (7, 6, 31) *       | law         | 0   | 2   | 2   | 2   | 2                | 2                | 2   | 2   | 4      | 3                        |
| Ternary random (7, 6, 31) *       | law         | 1   | 2   | 5   | 5   | 6                | 5                | 5   | 5   | 135    | 7                        |
| Thresholds ( $n = 7$ )            | depth $> 2$ | 0   | 3   | 3   | 3   | 3                | 3                | 3   | 3   | 7      | 7                        |
| Thresholds ( $n = 7$ )            | depth $> 2$ | 1   | 3   | 5   | 7   | 9                | 7                | 7   | 7   | 333    | 21                       |
| Thresholds ( $n = 15$ )           | depth $> 2$ | 0   | 4   | 3   | 4   | 4                | 4                | 4   | 4   | 15     | 15                       |
| Thresholds ( $n = 15$ )           | depth $> 2$ | 1   | 4   | 5   | 9   | 12               | 9                | 9   | 8   | 4,116  | 45                       |
| Parity ( $k = 3$ )                | depth $> 2$ | 0   | 3   | 3   | 3   | 3                | 3                | 3   | 3   | 7      | 3                        |
| Parity ( $k = 3$ )                | depth $> 2$ | 1   | 3   | 5   | 7   | 9                | 7                | 6   | 6   | 318    | 6                        |
| Parity ( $k = 4$ )                | depth $> 2$ | 0   | 4   | 3   | 4   | 4                | 4                | 4   | 4   | 15     | 4                        |
| Parity ( $k = 4$ )                | depth $> 2$ | 1   | 4   | 5   | 9   | 12               | 9                | 8   | 7   | 5,865  | 7                        |
| Point functions ( $n = 6$ )       | depth $> 2$ | 0   | 5   | 3   | 5   | 5                | 5                | 5   | 5   | 43     | 5                        |
| Point functions ( $n = 6$ )       | depth $> 2$ | 1   | 5   | 5   | 11  | 15               | 11               | 11  | 11  | 716    | 11                       |
| Point functions ( $n = 10$ )      | depth $> 2$ | 0   | 9   | 3   | 9   | 9                | 9                | 9   | 9   | 969    | 9                        |
| Point functions ( $n = 10$ )      | depth $> 2$ | 1   | 9   | 5   | 19  | 27               | 19               | 19  | 19  | 59,028 | 19                       |
| Boolean functions ( $b = 2$ )     | depth $> 2$ | 0   | 4   | 3   | 4   | 4                | 4                | 4   | 4   | 15     | 4                        |
| Boolean functions ( $b = 2$ )     | depth $> 2$ | 1   | 4   | 5   | 9   | 12               | 9                | 9   | 9   | 210    | 12                       |
| Monotone conjunctions ( $k = 3$ ) | depth $> 2$ | 0   | 3   | 3   | 3   | 3                | 3                | 3   | 3   | 7      | 3                        |
| Monotone conjunctions ( $k = 3$ ) | depth $> 2$ | 1   | 3   | 5   | 7   | 9                | 7                | 7   | 7   | 428    | 9                        |

**Table 13** The original supplementary 13-cell  $e = 2$  evaluation with shallow-rule results. It includes selected primary instances and additional small structured tables. “DAG nodes” is the size of the verified adversary certificate. For random tables, each triple gives  $(M, Q, \text{seed})$ . In the final row, the wall-clock cap is reached before an expanded proof is produced; the portable law still certifies  $\text{OPT}(2) = 8$ . For the ternary random table  $(7, 6, 31)$ , greedy returns 9 against the certified optimum 8, while the compact law already gives 8; here the symbolic certificate improves on the search-based heuristic.

| Instance                          | shallow result    | OPT(2)                                 | $R_{\text{dec}}$ | greedy | OPT <sup>na</sup> (2) | DAG nodes |
|-----------------------------------|-------------------|----------------------------------------|------------------|--------|-----------------------|-----------|
| Scalar DSL ( $r = 1$ )            | verified gap      | 9                                      | 11               | 10     | 14                    | 5,941     |
| Scalar DSL ( $r = 2$ )            | exact law, $3e+2$ | 8                                      | 8                | 8      | 10                    | 1,246     |
| Sequence DSL ( $d = 1$ )          | exact law, $2e+1$ | 5                                      | 5                | 5      | 5                     | 1         |
| Binary random $(5, 4, 7)$         | verified gap      | 11                                     | 11               | 11     | 15                    | 552       |
| Ternary random $(7, 6, 31)$       | exact law, $3e+2$ | 8                                      | 8                | 9      | 12                    | 247       |
| Thresholds ( $n = 5$ )            | verified gap      | 9                                      | 11               | 10     | 25                    | 439       |
| Thresholds ( $n = 7$ )            | verified gap      | 10                                     | 11               | 11     | 35                    | 2,708     |
| Parity ( $k = 2$ )                | exact law, $3e+2$ | 8                                      | 8                | 8      | 8                     | 70        |
| Parity ( $k = 3$ )                | verified gap      | 9                                      | 11               | 9      | 10                    | 617       |
| Point functions ( $n = 4$ )       | verified gap      | 11                                     | 11               | 11     | 11                    | 377       |
| Monotone conjunctions ( $k = 2$ ) | exact law, $3e+2$ | 8                                      | 8                | 8      | 10                    | 129       |
| Boolean functions ( $b = 1$ )     | exact law, $3e+2$ | 8                                      | 8                | 8      | 10                    | 15        |
| Sequence DSL ( $d = 2$ )          | exact law, $3e+2$ | 1800 s cap; no expanded proof produced |                  |        |                       |           |

## D Computational Checks

This appendix summarises the limits, acceptance criteria, and independent checks used for the computational results.

### D.1 Resource Limits and Completion

Every method had a declared cap. A cell that exceeded a cap retained the limiting stage but no numerical value for that method. Such cells remain in the denominators reported below.

Table 14 records the resource caps.

**Table 14** Per-method resource caps. The extended sweep at  $e = 2$  also used a 600-second wall-clock cap per cell; the depth-two sequence table in the supplementary suite used a specified 1800-second cap.

| Resource                    | Primary | Extended sweep<br>( $e \leq 1$ ) | $e = 2$<br>suites |
|-----------------------------|---------|----------------------------------|-------------------|
| Greedy robust states        | 250,000 | 1,000,000                        | 1,000,000         |
| Exact adaptive states       | 500,000 | 2,000,000                        | 2,000,000         |
| Adversary-certificate nodes | 500,000 | 2,000,000                        | 2,000,000         |
| Non-adaptive search nodes   | 500,000 | 2,000,000                        | 2,000,000         |

Table 15 distinguishes completed searches from combined certification.

**Table 15** Original per-method completion and combined adaptive certification after the isolation extension. “Any adaptive certificate” includes independently checked portable laws. The supplementary sequence value was already certified by a shallow law. Cells are declared evaluation cells; overlapping suites are not counts of distinct games.

| Suite                            | Cells | Expanded<br>adaptive | Any adaptive<br>certificate | Greedy | Non-<br>adaptive |
|----------------------------------|-------|----------------------|-----------------------------|--------|------------------|
| Primary, $e \in \{0, 1\}$        | 30    | 30                   | 30                          | 30     | 30               |
| Extended sweep, $e \in \{0, 1\}$ | 202   | 202                  | 202                         | 202    | 202              |
| Extended sweep, $e = 2$          | 101   | 98                   | 100                         | 101    | 100              |
| Supplementary suite, $e = 2$     | 13    | 12                   | 13                          | 12     | 12               |
| Total                            | 346   | 342                  | 345                         | 345    | 344              |

The original extended sweep at  $e = 2$  produced four method-level limits. The parity instance of dimension five exceeded the exact adaptive state cap and, separately, the non-adaptive node cap. The point-function instances with nine and ten classes each exceeded the adversary-certificate node cap. All 75 random-table proofs completed at every budget. In the supplementary suite, the depth-two sequence instance reached its 1800-second cap before an expanded proof was produced; its portable law still proves

$\text{OPT}(2) = 8$ . New isolation certificates independently prove 26 and 29 for the nine- and ten-point cells, respectively. None of these failed expanded runs is relabelled as complete; the combined-certification column records the separate proofs.

## D.2 Acceptance and Consistency Checks

A numerical result was retained only after its evidence passed an independent verifier. Compact claims required the upper strategy and lower witness, exact adaptive claims required both the strategy and adversary certificate, and non-adaptive claims required the code and rational dual. Whenever all relevant quantities existed, the checks included

$$\begin{aligned} L(e) &\leq \text{OPT}_A(e) \leq U(e), \\ \text{OPT}_A(e) &\leq \text{OPT}_A^{\text{na}}(e), \quad \text{OPT}_A(e) \leq \text{greedy depth}. \end{aligned}$$

A matched portable law also had to satisfy  $L = U = \text{OPT}$ , while a reported compact gap had to satisfy  $L < U$ . Every retained row passed these checks. The structural computations were deterministic: random families used fixed seeds, searches used fixed total orders, and no result depended on hash iteration order.

## References

- [1] Rényi, A.: On a problem of information theory. MTA Mat. Kut. Int. Kozl. **6B**, 505–516 (1961)
- [2] Ulam, S.M.: Adventures of a Mathematician. Charles Scribner’s Sons, New York (1976)
- [3] Rivest, R.L., Meyer, A.R., Kleitman, D.J., Winklmann, K., Spencer, J.: Coping with errors in binary search procedures. Journal of Computer and System Sciences **20**(3), 396–404 (1980) [https://doi.org/10.1016/0022-0000\(80\)90014-8](https://doi.org/10.1016/0022-0000(80)90014-8)
- [4] Spencer, J., Winkler, P.: Three thresholds for a liar. Combinatorics, Probability and Computing **1**(1), 81–93 (1992) <https://doi.org/10.1017/S0963548300000080>
- [5] Feige, U., Raghavan, P., Peleg, D., Upfal, E.: Computing with noisy information. SIAM Journal on Computing **23**(5), 1001–1018 (1994) <https://doi.org/10.1137/S0097539791195877>
- [6] Kenyon, C., Yao, A.: On evaluating boolean functions with unreliable tests. International Journal of Foundations of Computer Science **1**(1), 1–10 (1990) <https://doi.org/10.1142/S0129054190000023>
- [7] Maharjan, R., Watson, T.: Complexity of fault tolerant query complexity. In: 42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS). Leibniz International Proceedings in Informatics, vol. 250. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2022). <https://doi.org/10.4230/LIPIcs.FSTTCS.2022.26> . Pages 26:1–26:11

- [8] Angluin, D., Krikis, M., Sloan, R.H., Turán, G.: Malicious omissions and errors in answers to membership queries. *Machine Learning* **28**(2–3), 211–255 (1997) <https://doi.org/10.1023/A:1007311411259>
- [9] Nowak, R.D.: Noisy generalized binary search. In: *Advances in Neural Information Processing Systems 22 (NIPS)*, pp. 1366–1374 (2009)
- [10] Pelc, A.: Searching games with errors: Fifty years of coping with liars. *Theoretical Computer Science* **270**(1–2), 71–109 (2002) [https://doi.org/10.1016/S0304-3975\(01\)00303-6](https://doi.org/10.1016/S0304-3975(01)00303-6)
- [11] Cicalese, F.: *Fault-Tolerant Search Algorithms: Reliable Computation with Unreliable Information*. Monographs in Theoretical Computer Science. An EATCS Series. Springer, Berlin, Heidelberg (2013). <https://doi.org/10.1007/978-3-642-17327-1>
- [12] Garey, M.R.: Optimal binary identification procedures. *SIAM Journal on Applied Mathematics* **23**(2), 173–186 (1972) <https://doi.org/10.1137/0123019>
- [13] Laber, E.S., Nogueira, L.T.: On the hardness of the minimum height decision tree problem. *Discrete Applied Mathematics* **144**(1–2), 209–212 (2004) <https://doi.org/10.1016/j.dam.2004.06.002>
- [14] Bshouty, N.H., Makhoul, W.: On polynomial time constructions of minimum height decision tree. In: *29th International Symposium on Algorithms and Computation (ISAAC 2018)*. Leibniz International Proceedings in Informatics, vol. 123. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2018). <https://doi.org/10.4230/LIPIcs.ISAAC.2018.34> . Pages 34:1–34:12
- [15] McConnell, R.M., Mehlhorn, K., Näher, S., Schweitzer, P.: Certifying algorithms. *Computer Science Review* **5**(2), 119–161 (2011) <https://doi.org/10.1016/j.cosrev.2010.09.009>
- [16] Necula, G.C.: Proof-carrying code. In: *Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*, pp. 106–119. ACM, New York, NY, USA (1997). <https://doi.org/10.1145/263699.263712>
- [17] Appel, A.W.: Foundational proof-carrying code. In: *Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pp. 247–256 (2001). <https://doi.org/10.1109/LICS.2001.932501>
- [18] Wetzler, N., Heule, M.J.H., Hunt, W.A.: DRAT-trim: Efficient checking and trimming using expressive clausal proofs. In: *Theory and Applications of Satisfiability Testing (SAT)*. Lecture Notes in Computer Science, vol. 8561, pp. 422–429 (2014). [https://doi.org/10.1007/978-3-319-09284-3\\_31](https://doi.org/10.1007/978-3-319-09284-3_31)

- [19] Cheung, K.K.H., Gleixner, A., Steffy, D.E.: Verifying integer programming results. In: *Integer Programming and Combinatorial Optimization. Lecture Notes in Computer Science*, vol. 10328, pp. 148–160. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-59250-3\\_13](https://doi.org/10.1007/978-3-319-59250-3_13)
- [20] Applegate, D.L., Cook, W., Dash, S., Espinoza, D.G.: Exact solutions to linear programming problems. *Operations Research Letters* **35**(6), 693–699 (2007) <https://doi.org/10.1016/j.orl.2006.12.010>
- [21] Meng, K., Lin, C., Yang, Y.: Minimum number of queries for an adaptive liar search game with small sets. *Discrete Optimization* **10**(4), 233–240 (2013) <https://doi.org/10.1016/j.disopt.2013.07.002>
- [22] Fraknói, Á.X., Márton, D.Á., Simon, D.G., Lenger, D.A.: On the rényi–ulam game with restricted size queries. *Discrete Optimization* **48**, 100772 (2023) <https://doi.org/10.1016/j.disopt.2023.100772>
- [23] Angluin, D.: Queries and concept learning. *Machine Learning* **2**(4), 319–342 (1988) <https://doi.org/10.1023/A:1022821128753>
- [24] Guillory, A., Bilmes, J.: Simultaneous learning and covering with adversarial noise. In: *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pp. 369–376 (2011). [https://icml.cc/2011/papers/261\\_icmlpaper.pdf](https://icml.cc/2011/papers/261_icmlpaper.pdf)
- [25] Hamming, R.W.: Error detecting and error correcting codes. *The Bell System Technical Journal* **29**(2), 147–160 (1950) <https://doi.org/10.1002/j.1538-7305.1950.tb00463.x>
- [26] MacWilliams, F.J., Sloane, N.J.A.: *The Theory of Error-Correcting Codes*. North-Holland, Amsterdam (1977)
- [27] Dobson, G.: Worst-case analysis of greedy heuristics for integer programming with nonnegative data. *Mathematics of Operations Research* **7**(4), 515–531 (1982) <https://doi.org/10.1287/moor.7.4.515>
- [28] Chvátal, V.: A greedy heuristic for the set-covering problem. *Mathematics of Operations Research* **4**(3), 233–235 (1979) <https://doi.org/10.1287/moor.4.3.233>
- [29] Dinur, I., Steurer, D.: Analytical approach to parallel repetition. In: *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 624–633. ACM, New York, NY, USA (2014). <https://doi.org/10.1145/2591796.2591884>
- [30] Ellis, R.B., Nyman, K.L.: Two-batch liar games on a general bounded channel. *Journal of Combinatorial Theory, Series A* **116**(8), 1253–1270 (2009) <https://doi.org/10.1016/j.jcta.2009.03.005>
- [31] Berlekamp, E.R.: Block coding with noiseless feedback. PhD thesis, Massachusetts Institute of Technology (1964)

- [32] Dasgupta, S.: Analysis of a greedy active learning strategy. In: Advances in Neural Information Processing Systems 17 (NIPS), pp. 337–344 (2004)
- [33] Nowak, R.D.: The geometry of generalized binary search. *IEEE Transactions on Information Theory* **57**(12), 7893–7906 (2011) <https://doi.org/10.1109/TIT.2011.2169298>
- [34] Golovin, D., Krause, A.: Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research* **42**, 427–486 (2011) <https://doi.org/10.1613/jair.3278>
- [35] Balcan, M.-F., Beygelzimer, A., Langford, J.: Agnostic active learning. *Journal of Computer and System Sciences* **75**(1), 78–89 (2009) <https://doi.org/10.1016/j.jcss.2008.07.003>
- [36] Hanneke, S.: Theory of disagreement-based active learning. *Foundations and Trends in Machine Learning* **7**(2–3), 131–309 (2014) <https://doi.org/10.1561/2200000037>
- [37] Hanneke, S., Yang, L.: Minimax analysis of active learning. *Journal of Machine Learning Research* **16**(109), 3487–3602 (2015)
- [38] Hyafil, L., Rivest, R.L.: Constructing optimal binary decision trees is NP-complete. *Information Processing Letters* **5**(1), 15–17 (1976) [https://doi.org/10.1016/0020-0190\(76\)90095-8](https://doi.org/10.1016/0020-0190(76)90095-8)
- [39] Adler, M., Heeringa, B.: Approximating optimal binary decision trees. In: Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques. *Lecture Notes in Computer Science*, vol. 5171, pp. 1–9. Springer, Berlin, Heidelberg (2008). [https://doi.org/10.1007/978-3-540-85363-3\\_1](https://doi.org/10.1007/978-3-540-85363-3_1)
- [40] Chakaravarthy, V.T., Pandit, V., Roy, S., Awasthi, P., Mohania, M.K.: Decision trees for entity identification: Approximation algorithms and hardness results. *ACM Transactions on Algorithms* **7**(2) (2011) <https://doi.org/10.1145/1921659.1921661> . Pages 15:1–15:22
- [41] Moshkov, M.J.: Time complexity of decision trees. In: Transactions on Rough Sets III. *Lecture Notes in Computer Science*, vol. 3400, pp. 244–459. Springer, Berlin, Heidelberg (2005). [https://doi.org/10.1007/11427834\\_12](https://doi.org/10.1007/11427834_12)
- [42] Narodytska, N., Ignatiev, A., Pereira, F.S., Marques-Silva, J.: Learning optimal decision trees with SAT. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, pp. 1362–1368 (2018). <https://doi.org/10.24963/ijcai.2018/189>
- [43] Blum, M., Kannan, S.: Designing programs that check their work. *Journal of the ACM* **42**(1), 269–291 (1995) <https://doi.org/10.1145/200836.200880>

- [44] Blum, M., Luby, M., Rubinfeld, R.: Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences* **47**(3), 549–595 (1993) [https://doi.org/10.1016/0022-0000\(93\)90044-W](https://doi.org/10.1016/0022-0000(93)90044-W)
- [45] Katz, G., Barrett, C., Dill, D.L., Julian, K., Kochenderfer, M.J.: Reluplex: An efficient SMT solver for verifying deep neural networks. In: *Computer Aided Verification (CAV)*. *Lecture Notes in Computer Science*, vol. 10426, pp. 97–117 (2017). [https://doi.org/10.1007/978-3-319-63387-9\\_5](https://doi.org/10.1007/978-3-319-63387-9_5)
- [46] Cohen, J.M., Rosenfeld, E., Kolter, J.Z.: Certified adversarial robustness via randomized smoothing. In: *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 1310–1320 (2019)
- [47] Blanc, G., Koch, C., Lange, J., Tan, L.-Y.: The query complexity of certification. In: *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pp. 623–636. ACM, New York, NY, USA (2022). <https://doi.org/10.1145/3519935.3519993>
- [48] Bogaerts, B., Gocht, S., McCreesh, C., Nordström, J.: Certified dominance and symmetry breaking for combinatorial optimisation. *Journal of Artificial Intelligence Research* **77**, 1539–1589 (2023) <https://doi.org/10.1613/jair.1.14296>
- [49] Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W. (eds.) *Complexity of Computer Computations*, pp. 85–103. Plenum Press, New York (1972). [https://doi.org/10.1007/978-1-4684-2001-2\\_9](https://doi.org/10.1007/978-1-4684-2001-2_9)

## Supplementary Files

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

- [ESM1.zip](#)