AUDITING DISCOVERY CLAIMS: A TWO-SIDED CRITERION FOR AGENTIC SCIENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

When an agentic science system improves against a fallible evaluator, what establishes a new capability rather than evaluator-specific optimization? We propose a two-sided audit that separates an exact exclusion for the prior verifier from empirical gains under a shared adjudicator. RNA topology supplies the negative side: a pseudoknot-free verifier cannot certify crossing structures at any search budget. The positive side is model-relative, assessed by three structure predictors with one held outside every optimization loop. The audit returns six findings rather than a pass/fail bit. A hand-built operator solves 43 of 60 crossing targets under the predictor it optimizes, but only one under all three. On the same 43 targets, an unoptimized predictor confirms two of its designs versus 26 for an external solver. Auditing our revised operator finds a narrow improvement but no measured advantage over matched-cap undirected search. The instrument nevertheless discriminates: two failure-free frozen LLM-written operators attain held-out carry-over of 0.293 versus 0.095 on 951 paired units, using 4.6–10 times fewer oracle calls. Seven interventional tests do not explain the difference. These results support externally adjudicated, compute-accounted capability comparisons, not physical validation, a policy-level grammar transition, or discovery of a transferable design principle.

1 INTRODUCTION

A scientific agent may report progress because its benchmark score, reward, or fit improves. Such evidence is useful but incomplete when the system also searches against the evaluator that supplies the score. More search, changes in what the verifier can express, and optimization of a surrogate's blind spots are different explanations for the same apparent gain. An audit should report which alternatives it excludes, rather than treating a scalar improvement as a discovery certificate.

We separate two questions: what could the prior verifier certify, and what can the revised system realize under a shared adjudicator? RNA inverse design makes the first question unusually precise. Nested secondary structures admit context-free descriptions; selected unbounded pseudoknot families require greater generative power (Rivas & Eddy, 2000; Nebel & Weinberg, 2012; Kato et al., 2006). A pseudoknot-free folding oracle never returns a crossing structure, regardless of the policy or search budget. This is an exact restriction on the *verifier*, not an impossibility theorem about the prior policy or the molecule. The positive question has no comparable exactness: computational structure predictors disagree, and agreement is not experimental confirmation.

Our contribution is an evidence vector that keeps these epistemic levels separate. We instantiate it with a panel of pKiss, ProbKnot, and ShapeKnots, seal candidates before external adjudication, and compare prior, single-oracle, undirected, and solver-free arms. The resulting evidence is deliberately asymmetric. The headline $43/60 \rightarrow 1/60$ collapse diagnoses evaluator dependence; deterministic ablations show that our own objective provides no measured panel-confirmed advantage over undirected search. A positive control prevents interpreting those low rates as an entirely blind instrument: two frozen LLM-written procedures exhibit a larger held-out gain with fewer realized oracle calls. Their mechanism remains unidentified after seven interventions.

We contribute (i) a formal object for verifier-range exclusion, (ii) a two-sided reporting protocol, and (iii) a controlled empirical audit including its negative findings. We do not propose a new

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

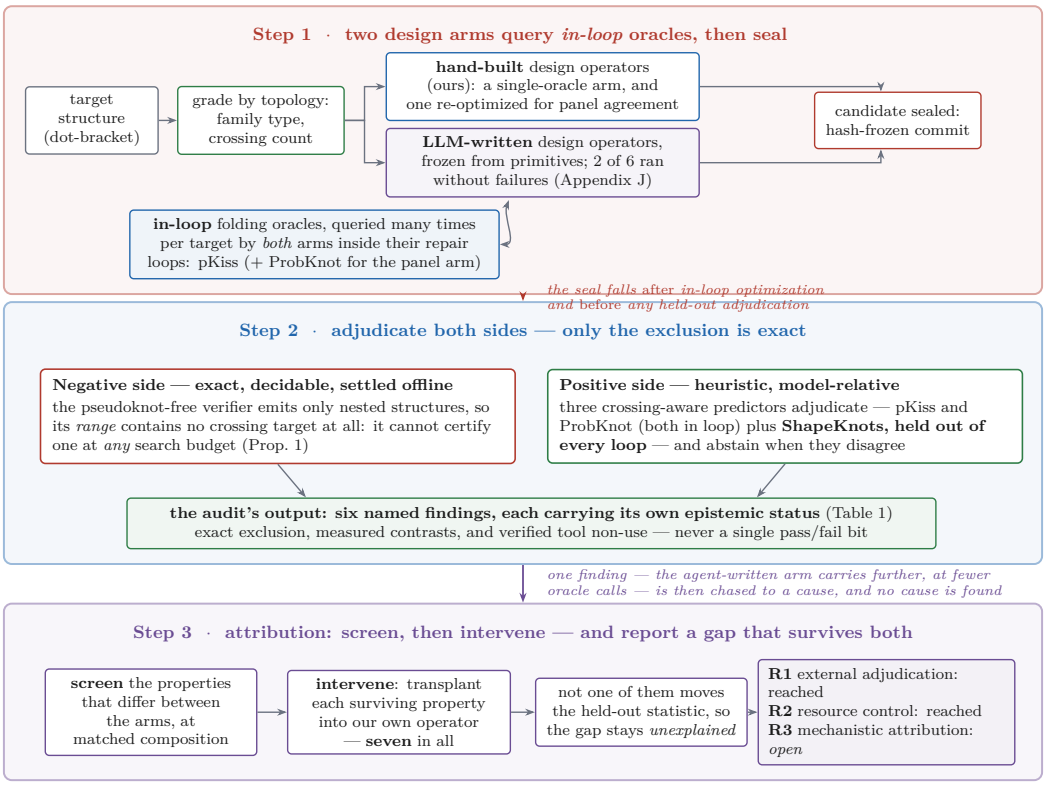

Figure 1: **A two-sided audit, followed by a separate attribution test.** The old verifier's range restriction is exact; positive findings remain predictor-relative. Candidates are sealed before ShapeKnots adjudication, not before in-loop optimization. The six-component verdict distinguishes measured contrasts from verified tool non-use. R1 and R2 describe the two selected failure-free agent-written operators; no transferable mechanism (R3) is established.

folding model or establish autonomous scientific discovery. Detailed definitions, proofs, alternative estimators, and the extended experimental record appear after the references.

Figure 1 outlines the evidence flow. Keeping the negative and positive sides separate prevents a change in what an evaluator can express from being mistaken for a change in what a policy knows. Keeping attribution separate prevents a robust difference between policies from being mistaken for an explanation of that difference.

2 What the Negative Side Can Prove

Let $T = (n, P)$ denote a structure with base-pair set P. A crossing exists if $(i, j), (k, \ell) \in P$ satisfy $i < k < j < \ell$. Let $\mathcal{L}_{\text{nest}}$ contain nested structures and $\mathcal{L}_{\text{pk}}$ contain crossing structures. A folding oracle Ω maps sequences to single structures. For a target-conditioned policy π, define its realizable-target language as

$$\mathcal{R}_\Omega(\pi) = \{T : \Pr_{x \sim \pi(\cdot|T)}[\Omega(x) = T] > 0\}. \tag{1}$$

It is the intersection of what the oracle can realize and what the policy reaches; in particular $\mathcal{R}_\Omega(\pi) \subseteq \operatorname{im}\Omega$.

Proposition 1 (Verifier-range exclusion)**.** *If $\operatorname{im}\Omega_{\text{cf}} \subseteq \mathcal{L}_{\text{nest}}$, then $\mathcal{R}_{\Omega_{\text{cf}}}(\pi) \cap \mathcal{L}_{\text{pk}} = \varnothing$ for every policy π. For a given target, a crossing witness can be checked in $O(|P|^2)$, independently of any design search budget.*

Proof. Every realized target must be an oracle output. Those outputs are nested by assumption, whereas a crossing witness excludes membership in $\mathcal{L}_{\text{nest}}$. Enumerating pairs of base pairs checks for that witness. □

This elementary range argument is the exact part of the audit. It does not require proving the computational capacity of the agent, and it holds even for an arbitrarily powerful sequence generator. The Chomsky framing contributes an externally specified family axis, not a stronger theorem about finite observations. An H-type family with two arbitrarily long interleaving stems is non-context-free (Appendix F); observing finitely many realized members cannot demonstrate a policy-level class transition, because every finite language is regular. Crossing-stem count is an intrinsic structural grade, not a grammar dimension. Generative power must also be distinguished from folding complexity (Lyngsø & Pedersen, 2000; Bonnet et al., 2020).

Two impossibilities with different quantifiers. Verifier confinement fixes an oracle and excludes a target for *every policy*; undesignability fixes a target and energy criterion and excludes *every sequence*. A rival-structure certificate can establish the latter by exhibiting a competitor at least as favorable as the target for every compatible sequence, preventing the target from being the unique optimum (Zhou et al., 2024; 2025). Such a certificate binds the revised system too: changing the policy cannot overcome impossibility under the same criterion. In contrast, range confinement can disappear when the verifier changes, without establishing that the policy learned anything. This is why the exact range theorem must be paired with an empirical prior-policy floor under the shared adjudicator, rather than counted as evidence of learning by itself.

We exclude targets only on exact rival-structure certificates, never on failed search. No primary-pool target is excluded. Consequently the oracle-relative designability of unsolved targets remains unknown: native provenance does not guarantee that a computational model can recover their structure. Abstention therefore reports a lack of positive evidence, not a certificate that the target is impossible.

3 A Two-Sided Audit

Acceptance and held-out information. For predictor j, let $A_j(x, T) = \mathbf{1}[P(\Omega_j(x)) = P(T)]$. Exact equality refers to the entire pair set, not its nested projection. The revised hand-built operator searches against pKiss and ProbKnot; ShapeKnots is outside every arm's design and invention loop. We distinguish three rates:

$$r_{\text{in}} = \widehat{\mathbb{E}}[A_{\text{pKiss}} A_{\text{ProbKnot}}], \quad r_{\text{cons}} = \widehat{\mathbb{E}}[A_{\text{pKiss}} A_{\text{ProbKnot}} A_{\text{ShapeKnots}}], \quad r_{\text{ho}} = \widehat{\mathbb{E}}[A_{\text{ShapeKnots}}]. \tag{2}$$

The panel abstains on disagreement; abstentions remain in the attempted-target denominator and earn no success credit. These rates must not be blended into an unspecified "panel score." Conditional carry-over C instead asks what fraction of an arm's in-loop successes the held-out predictor confirms. For paired comparisons we explicitly name the intersection of successes used as its denominator.

Candidates and their target commitments are fixed before held-out adjudication. Sealing does not prevent in-loop overfitting: it prevents selecting it away with the held-out predictor, making collapse observable. The three predictors are distinct, not independent. pKiss and ShapeKnots share substantial assumptions, and ShapeKnots is run de novo without probing constraints, outside its strongest validation setting (Janssen & Giegerich, 2015; Bellaousov & Mathews, 2010; Hajdin et al., 2013).

Six outputs, not a discovery bit. Write a, b, s, u for after, before, single-oracle, and undirected policies. The audit returns

$$\mathcal{A} = (E_{\text{range}}, \Delta_{\text{prior}}, \Delta_{\text{single}}, \Delta_{\text{undirected}}, \Delta_{\text{held-out}}, g_{\text{aut}}), \tag{3}$$

where E_{range} is Proposition 1's exclusion, $\Delta_{\text{prior}} = r^a_{\text{cons}} - r^b_{\text{cons}}$, $\Delta_{\text{single}} = r^a_{\text{cons}} - r^s_{\text{cons}}$, $\Delta_{\text{undirected}} = r^a_{\text{cons}} - r^u_{\text{cons}}$, and $\Delta_{\text{held-out}} = r^a_{\text{ho}} - r^s_{\text{ho}}$. Finally, g_{aut} records the verified-rate change on removing the declared packaged inverse-design solver class. It measures dependence on that tool class, not freedom from folding oracles or all inverse-folding primitives. In this study packaged pseudoknot inverse-design solvers are never on the revised operator's path, so this component has no counterfactual leverage.

Table 1: Audit of the revised hand-built operator. Values are model-relative; intervals are 95%. The prior row is a range of execution-level estimates, not a confidence interval.

Component	Finding and scope
E_{range}	Exact verifier-range exclusion; not a bound on policy capacity.
Δ_{prior}	Prior $0/60$ in 24 executions; revised 1–$9/60$ (mean 3.29). Different optimization objectives limit the comparison.
Δ_{single}	Narrow: deterministic $+0.067$ $[+0.023, +0.120]$; target-set and realized-use caveats remain.
$\Delta_{\text{undirected}}$	Not established: $+0.000$ $[-0.060, +0.050]$ on 300 paired deterministic cells.
$\Delta_{\text{held-out}}$	Weak: direction positive at each tested seed, significant in only one of three.
g_{aut}	Verified non-use of packaged pseudoknot solvers; not a causal attribution of the gain.

Table 2: **Target pools and the claims they permit.** Identity holdout is not structural holdout. The 27-target pool is a structure-filtered subset, not an additional independent corpus. The separate 80-native calibration set assesses the predictors rather than design success.

Pool	n	Contact with development / interpretation
Primary development/evaluation	60	Hand-built operator and hyperparameters developed here.
Inspected secondary	50	Inspected, not tuned on; same-regime replication.
Invention training	16	Feedback for selecting the frozen LLM-written code.
ID-disjoint follow-up	125	No development feedback; 39 exact structural duplicates of touched targets.
Structurally disjoint	27	24 clean clusters; every target more than 15% from touched structures.
Cross-source design (bpRNA)	60	Different source and harder pool; not difficulty-matched.

Only the range exclusion is a theorem. The rate contrasts use the same targets and adjudicator and require uncertainty estimates; the prior-policy floor remains a finite-budget measurement even when it is zero at every execution. Target-level dependence, execution variation, realized compute, and runtime failures must remain visible. A conjunction of six pass/fail gates would erase these distinctions and could mistake one lucky target for a general capability.

Evidence levels. We separately classify the agent-written result: R1 is an advantage under an evaluator outside the optimization loop; R2 adds realized-compute controls; R3 requires an identified, transferable design principle. These are evidence levels, not additional vector components. No arm in this paper reaches R3.

4 Experiments

Setting and scope. The primary pool comprises 60 Pseudobase++ crossing targets. It is a *development/evaluation pool*: the hand-built operator and hyperparameters were developed on it. Target holdout for the frozen invention loop does not make this pool held out for every arm. ViennaRNA 2.7.2 supplies pseudoknot-free primitives, pKiss 2.3.0 the reference predictor, and RNAstructure 6.6 ProbKnot and ShapeKnots. The hand-built method combines nested decomposition and stochastic repair; the single-oracle baseline optimizes pKiss alone. DesiRNA is an external minimum-free-energy inverse-design comparator. Full configurations and protocols are in Appendices G and K.

Experimental comparisons. Table 2 separates three notions often conflated by "held out": an evaluator excluded from optimization, target identities excluded from feedback, and structural families excluded from development. ShapeKnots satisfies the first; the primary pool does not satisfy the latter two for the hand-built arm. The frozen invention protocol uses a separate 16-target training pool, but this does not retroactively make its comparator's development targets unseen.

Units and uncertainty. The wall-clock and deterministic studies answer related but distinct questions. Repeated executions expose runtime variation; counted allowances and explicit random-generator seeding test reproducibility. Paired analyses compare methods on the same target and

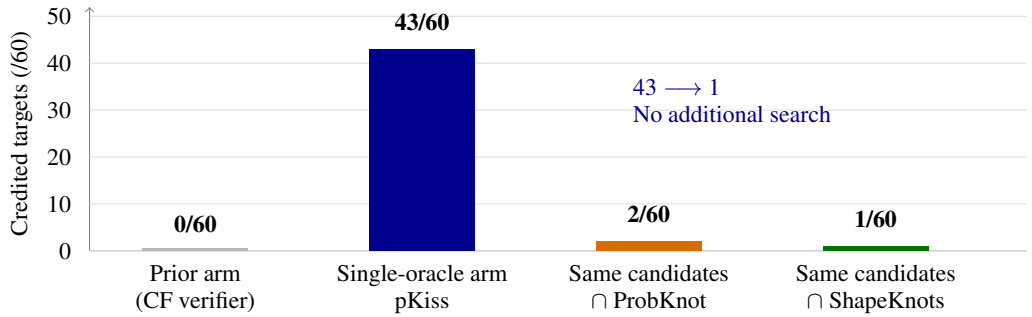

Figure 2: **The same candidates lose credit under additional predictors.** The single-oracle hand-built arm obtains 43 pKiss matches, 2 also confirmed by ProbKnot, and 1 confirmed by all three, out of 60 targets. The prior arm is at 0/60. Additional tests adjudicate sealed candidates; they do not rerun search. Only ShapeKnots is excluded from every arm's optimization loop.

Table 3: Same 43 pKiss-solved targets, different carry-over. ProbKnot is unoptimized by these two arms; it is not held out from the revised two-oracle arms elsewhere in the paper.

Method	pKiss	ProbKnot	ShapeKnots	All three
DesiRNA (MFE)	43/43	26/43	1/43	0/43
Hand-built (pKiss)	43/43	2/43	3/43	1/43

execution cell, with clustering at the target level so that repeated successes on one easy target do not masquerade as independent evidence. Conditional carry-over is additionally paired on the intersection of in-loop successes. That estimand controls target composition within the selected subset, but does not estimate unconditional performance on a fresh population. We therefore report both its denominator and the unconditional all-three metric.

4.1 Single-oracle success does not imply carry-over

Figure 2 visualizes the headline discrepancy without treating agreement as physical truth. The single-oracle operator attains $43/60$ exact pKiss matches, but only $1/60$ under all three predictors. A large ratio alone is fragile when its denominator is one. The stronger evidence is a paired cross-predictor comparison with DesiRNA: the two methods solve the *same* 43 pKiss targets (Jaccard 1.000), yet ProbKnot confirms 2 of our designs and 26 of DesiRNA's (Table 3). The discordant counts are 25 versus 1, exact McNemar $p = 8.1 \times 10^{-7}$, paired risk difference $+0.558$ $[+0.395, +0.721]$. At the full 60-target denominator the difference is $+0.400$ $[+0.267, +0.533]$.

Matching targets rules out target composition, not other differences between the methods. Search algorithm, energy model, candidate quality, and budget remain confounded. Thus the contrast is consistent with stronger oracle-specific optimization, not a randomized causal localization to the training oracle. Neither method is robust under all three predictors, and superiority under ProbKnot reverses under ShapeKnots. Cross-predictor robustness is pairwise, not a guarantee of correctness.

4.2 The audit is mostly negative on our own operator

Table 4 separates the benefit of running a repair loop from the benefit of guiding it with the proposed objective. Re-optimizing with the pKiss–ProbKnot conjunction recovers a small set of all-three-confirmed targets, but replication reduces the estimated advantage over single-oracle search: $+0.080 \rightarrow +0.056 \rightarrow +0.048$ in successive wall-clock analyses. Fixed Python seeds were insufficient because ViennaRNA maintains a separate global random generator and wall-clock budgets vary with load. The deterministic regime seeds both and replaces time caps with counted allowances. Five seeds and 60 targets then give $\Delta_{\text{single}} = +0.067$ $[+0.023, +0.120]$, with the single-oracle baseline at $0/60$ for every seed (Appendix Table 12).

Table 4: **Deterministic ablations at a shared 150-call pKiss allowance.** Means over five seeds, 60 targets each. In-loop predicates differ for the single-oracle arm, so those counts are not a shared outcome. Δ is full minus ablated all-three rate; intervals bootstrap targets and seeds. Actual calls differ because of stopping.

Arm	In-loop /60	All three /60	Δ [95%]	Calls
Single-oracle	42.6	0.00	$+.067$ [$+.023, +.120$]	52.6
Full (reference)	37.0	4.00	—	82.2
No objective (undirected)	28.6	4.00	$+.000$ [$-.060, +.050$]	99.5
No objective or bias	25.4	2.40	$+.027$ [$-.017, +.077$]	105.5
No repair loop	1.0	0.00	$+.067$ [$+.023, +.123$]	1.0
No CF decomposition	31.4	3.40	$+.010$ [$-.033, +.050$]	100.1
No composition step	34.4	5.40	$-.023$ [$-.080, +.030$]	90.3

Table 5: Paired held-out carry-over on units where both arms meet the same in-loop bar. Hand-built rates change because matched subsets differ. Intervals resample targets. Calls are realized pKiss calls per target in the same 20-second regime; hand-built reference: 128.0.

Operator	Matched n	Agent / hand-built	Paired difference [95%]	Calls
Gemini-3.1-pro	442	0.364 / 0.102	$+0.262$ [$+0.151, +0.382$]	27.8
Kimi-k3	509	0.232 / 0.088	$+0.143$ [$+0.059, +0.241$]	13.1
Pooled	951	0.293 / 0.095	$+0.199$ [$+0.108, +0.297$]	—

Within the same revised operator, removing objective guidance changes the in-loop rate by $+0.140$ $[+0.063, +0.220]$ but the all-three rate by exactly zero: $[-0.060, +0.050]$ on 300 paired cells. This is a null point estimate with nonzero uncertainty, not proof of equivalence. Removing the repair loop reduces the mean in-loop count from 37.0 to 1.0 and all-three confirmation to zero. Nested decomposition has an in-loop contribution of $+0.093$ $[+0.023, +0.173]$. The supported description is stochastic repair with no demonstrated held-out advantage from its objective guidance.

All deterministic arms receive a 150-call pKiss allowance, but use 52.6–105.5 calls on average because successful runs stop early. Equal caps are not equal realized work. The single-versus-panel comparison also changes more than guidance, so it cannot by itself establish that a changed success predicate caused the improvement.

A search-free compatibility probe further bounds interpretation. Sixty random compatible samples hit 30 of the operator's 43 pKiss-solved targets exactly. Across replicated comparisons, 84% of the panel-versus-single-oracle advantage lies on this zero-margin stratum; the remaining stratum contributes a $+0.016$ difference. This locates the result on targets that are easy for a finite random probe, not on an oracle-independent designability frontier.

4.3 A POSITIVE CONTROL WITH AN UNEXPLAINED ADVANTAGE

An audit that rejects almost everything may simply be insensitive. Frozen LLM-written operators supply a positive control under the same pKiss–ProbKnot acceptance predicate, with ShapeKnots outside both the invention and execution loops. The invention protocol provides primitives rather than a recipe, scores proposed code on 16 training targets for up to six rounds, and freezes the selected code before external adjudication (Appendix J). Only two of six operators run without failures on the primary pool; the following claim is about those two, not LLM-written operators generally.

We re-execute frozen code at five seeds and three executions per seed. On the 951 units where both the agent-written and hand-built arms satisfy their shared in-loop predicate, ShapeKnots confirms $279/951 = 0.293$ versus $90/951 = 0.095$. Discordant pairs favor agents by 218 to 29. These are repeated observations of targets, not 951 independent trials: target-clustered uncertainty gives $+0.199$ $[+0.108, +0.297]$ and target-cluster permutation $p = 5 \times 10^{-5}$. The exact cell-level McNemar would substantially overstate precision. Each model separately reproduces the direction (Table 5).

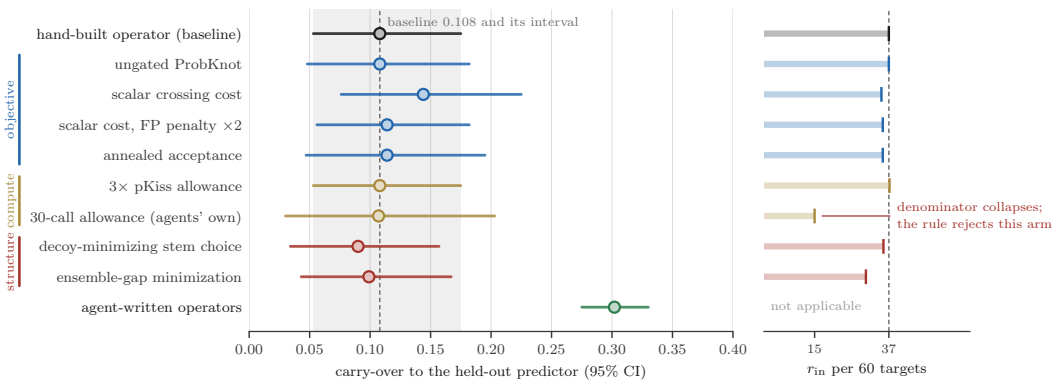

Figure 3: **A positive control does not supply its own explanation.** Seven mechanism tests and two compute controls are shown against the hand-built baseline and agent-written reference. Rates here condition on each arm's own successes, unlike Table 5's paired estimand. Error bars resample targets (95%); the right panel exposes denominator loss. This descriptive plot motivates, but does not replace, paired tests: a higher conditional rate bought by solving fewer targets is not a recovered capability.

Compute and mechanism are different questions. Figure 3 separates a robust outcome difference from the interventions intended to reproduce it. Source-code loop bounds suggest the agents search harder; realized calls show the opposite. They use 4.6–10 times fewer pKiss calls per target than the hand-built reference. An expanded-compute hand-built control reaches 196.2 calls without explaining the advantage. This is evidence for R2 under the reported compute accounting, not equality of every resource cost.

Seven candidate mechanisms were transplanted into the hand-built procedure and failed to reproduce the held-out gain. The protocol first screens properties under composition-adjusted controls, then verifies that an intervention changes its intended property before measuring carry-over. Two especially plausible surviving properties illustrate why screening is insufficient: a decoy-related feature differs between arms and predicts success within the hand-built arm, yet biasing toward it changes carry-over from 0.108 to 0.107; an ensemble-gap feature differs at matched composition, yet its transplant gives an exactly zero paired difference. Negative controls also exposed composition confounds in earlier distributional tests. These results reject the tested interventions as explanations; they do not establish that the properties can never matter.

The advantage is not confined to targets reached by the compatibility probe: on the 30 positive-margin targets, the two agent-written operators obtain 45 and 54 wins against the hand-built operator's 7. Nevertheless, failures on a structurally disjoint pool filter the denominator, and four of six original operators time out on 19–32 primary targets. Thus the primary-pool conditional advantage supports R1 and compute-accounted R2, while reliable transfer and a transferable mechanism (R3) remain open.

What remains between R2 and R3. A paired advantage is a between-system association, not an algorithmic principle. Realized-call controls exclude one resource explanation; attribution additionally requires manipulating a proposed property and recovering the held-out gain without sacrificing the success denominator. Failed transplants leave R1/R2 intact but block R3. "New" is relative to the prior system, not an unauditable claim about the base model's latent competence.

The positive control is integral to the audit: rejection alone cannot distinguish stringency from insensitivity. The selected operators demonstrate discrimination under a common acceptance rule, while their failed transplants show why discrimination is not explanation.

4.4 Target splits and the adjudicator's ceiling

Figure 4 supplies the reference class needed to interpret a low confirmation rate. Identity-disjoint evaluation is not structural transfer. Among 125 ID-disjoint targets, 39 are exact structural dupli-

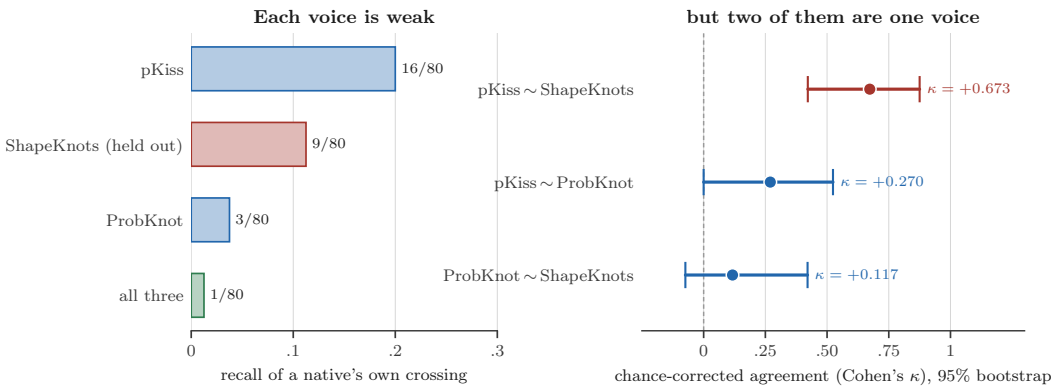

Figure 4: **The adjudicator has low recall and correlated errors.** On 80 native bpRNA pseudoknots, individual crossing recovery is 16/80, 9/80, and 3/80; unanimity is 1/80. Chance-corrected pairwise agreement (right; 95% bootstrap intervals) reveals substantial pKiss–ShapeKnots correlation. The positive paired comparison therefore establishes discrimination within this predictor panel, not independent confirmation of molecular correctness.

cates of touched targets and 94 lie within 15% normalized dot-bracket distance. Clustering all 251 crossing targets and excluding every cluster containing touched targets leaves 27 targets in 24 clean clusters. This is too small for precise low-base-rate transfer estimates. The hand-built revised operator exceeds the single-oracle baseline in all 15 repeated execution cells on this split, but repeated cells are not independent new targets. Appendix I records the split and its limitations.

More power is not structural generalization. The parameterized family $\mathcal{L}_{\mathrm{H}}$ permits a larger controlled test where the natural corpus cannot supply independent clusters. On the deterministic grid $n, m \in \{3, \ldots, 12\}$ (100 targets, five seeds), the single-oracle operator reaches $500/500$ in-loop matches but only $9/500$ all-three confirmations; the panel operator reaches $167/500$. Pairing within target and seed and clustering on target gives $\Delta_{\mathrm{single}} = +0.316\ [+0.258, +0.378]$ (Appendix Figure 14). This is stronger evidence for an effect on this family, not an estimate transferable to Pseudobase++. Its uniform topology is aligned with the operator's scaffold; additional samples improve precision without creating a second structural regime.

The size trend further separates procedure performance from evaluator behavior. All-three confirmation rises from 0.186 for $n + m \leq 12$ to 0.529 for $n + m \geq 18$, yet in-loop success falls from 0.900 to 0.736. Conditional carry-over rises from 0.206 to 0.718: the confirmed rate is the product of in-loop success and carry-over, and the second factor masks deterioration in the first. The single-oracle arm's pKiss-to-ProbKnot agreement instead falls from 0.093 to 0.036, ruling out a blanket increase in predictor agreement with size. The supported observation is no cliff in confirmed performance over the tested range, not improved search reach or unbounded generalization. Reporting only the final confirmed rate would conceal the operator's size-related limitation.

On 80 native bpRNA pseudoknots, pKiss, ShapeKnots, and ProbKnot recover known crossings for 16, 9, and 3 sequences, respectively; all three agree on only one. The held-out predictor's low recall limits what absence of confirmation means. Moreover pKiss–ShapeKnots agreement has Cohen's $\kappa = 0.673\ [0.422, 0.875]$, whereas the two ProbKnot pairings are less well determined. Unanimity is not three independent confirmations. The agent-written contrast demonstrates discrimination within this panel's narrow reference class, not high recall or physical correctness.

What would overturn the positive comparison. It is an association on the pool the operators were developed against, and it would fall if a fourth predictor from a different model class withdrew the held-out confirmations, or if a structurally independent corpus dropped the contrast; neither is available in this environment. The resource explanation is the one control the benchmark does allow: an arm held to the agents' own realized oracle volume carries over at 0.107 where they reach 0.367, so realized call count alone does not explain the difference. What the difference *is* remains the open problem this paper hands on.

5 Related Work

Agentic science systems automate proposal, execution, and revision (Gottweis et al., 2025; Wei et al., 2025); self-improving agents extend their skill libraries (Wang et al., 2023). Compression-based discovery criteria (Grünwald, 2007; Wang & Buehler, 2026) address fit and economical revision. These are different questions from a prior verifier's range exclusion or carry-over to an unoptimized evaluator: an information criterion alone provides neither R1 nor the realized-compute control of R2. We compare evidence, not the ambition of these systems.

The 42-target gap concerns external adjudication, not a direct test of any compression gate: in-loop fit and description length alone cannot establish agreement with an unqueried evaluator.

Reward hacking and proxy over-optimization are established problems (Amodei et al., 2016; Skalse et al., 2022; Gao et al., 2023). Falsification-oriented scientific agents seek stronger tests of proposed findings (Huang et al., 2025; Liu et al., 2024), while exact-evaluator construction search can demonstrate improvements under a fixed objective (Romera-Paredes et al., 2024; Fawzi et al., 2022; Mankowitz et al., 2023). Our difference is to combine a decidable restriction on the old verifier with explicitly fallible positive adjudication and matched comparisons. We do not claim to introduce held-out validation: independent-model self-consistency is already used in molecular design (Anishchenko et al., 2021; Watson et al., 2023). The contribution is the audit's separation of epistemic levels and the measured discrepancy it exposes.

Formal-language benchmarks characterize architectural generalization (Delétang et al., 2023); automaton accounts of agents characterize computational substrate (Koohestani et al., 2025). Here the language class belongs to a target family, not the neural model. We invoke established RNA formal-language results and reuse a related verifier study (Anonymous, 2026), rather than presenting either as a newly discovered biological law.

6 Discussion and Conclusion

The contribution is an audit of evidence, not inverse-design state of the art. It exposes the $43/60 \to 1/60$ evaluator dependence while withholding credit for our own objective guidance. The two selected LLM-written operators demonstrate a paired advantage at lower oracle use, not a transferable mechanism: seven transplants fail to recover their gain. Thus a discriminating instrument can support a positive comparison without endorsing its proposed explanation.

Limitations and reproducibility boundaries. The natural corpus supplies only 24 clean structural clusters; the synthetic family improves power, not structural diversity. The panel shares nearest-neighbour assumptions and has low native recall (Figure 4); ShapeKnots is used without probing data, and no wet-lab confirmation is attempted. Most historical operators were not bit-reproducible: the ablation and headline comparison were rerun deterministically, but pool-level and cross-source results were not. Frozen-source re-execution reproduces the selected procedures, not their invention through routed aliases without model-version hashes. We hold no timestamped pre-registration; estimator choices remain author-reported degrees of freedom. Equal oracle allowances are not equal realized costs, and repeated executions do not create independent targets. These are distinct limitations: more runs cannot replace an independent corpus, a different predictor family, or an independently replicated invention step.

Portability requires more than an exact evaluator. Exact positive verification, as in proof assistants or SAT checkers, does not itself characterize a prior verifier's range. Without a demonstrable range restriction, E_{range} has no counterpart: the negative side becomes an empirical floor. Held-out adjudication, matched comparisons, and compute accounting still transfer, but are not decidable exclusion. Revision and audit are complementary: fit can motivate a revision; external adjudication tests its benefit. The next decisive tests require an independent corpus, a different predictor family, and a successful mechanism intervention. Neither higher surrogate scores nor unanimity licenses physical discovery.

AI USE STATEMENT

AI assistance was used to condense and draft this submission from the full manuscript and to organize submission materials. Separately, the research evaluates frozen LLM-written operators; their generation, selection, runtime failures, and evaluation are documented in Appendix J. The models are experimental subjects, not independent validators of the scientific claims. The authors retain responsibility for checking the manuscript, citations, analyses, and any AI-assisted content.

ETHICS STATEMENT

This study audits computational evidence for capability claims using RNA structure prediction. It does not establish experimentally validated molecular function, safety, or suitability for biological deployment. Low predictor recall and correlated errors make that distinction important. No human-subject study or personal-data analysis is reported. Any reuse or redistribution of third-party datasets and prediction tools must respect their licenses. We recommend treating the reported designs and scores as computational audit artifacts, not as experimentally certified molecules.

REPRODUCIBILITY STATEMENT

The appendix supplies complete formal arguments (Appendix F), protocol and implementation details (Appendices G and K), and target-split limitations (Appendix I). The accompanying supplementary archive contains recorded results and offline analysis scripts, with an inventory and checksums. These support re-analysis of the reported measurements without new model calls. Re-executing designs additionally requires external folding tools and datasets; wall-clock experiments and third-party services need not reproduce bit for bit. No new scientific experiments were performed for this condensed submission.

REFERENCES

Jacobo Aguirre, Javier M. Buldú, Michael Stich, and Susanna C. Manrubia. Topological structure of the space of phenotypes: The case of RNA neutral networks. *PLoS ONE*, 6(10):e26324, 2011. doi: 10.1371/journal.pone.0026324.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Ivan Anishchenko, Samuel J. Pellock, Tamuka M. Chidyausiku, et al. De novo protein design by deep network hallucination. *Nature*, 600:547–552, 2021.

Anonymous. "solved" is a choice of verifier, not a fact: Deterministic cost–robustness frontiers of verifier choice in RNA and SAT. Technical report, Anonymous, 2026. companion technical report; author and institution withheld for double-blind review.

Stanislav Bellaousov and David H. Mathews. ProbKnot: Fast prediction of RNA secondary structure including pseudoknots. *RNA*, 16(10):1870–1880, 2010. doi: 10.1261/rna.2125310.

Édouard Bonnet, Paweł Rzążewski, and Florian Sikora. Designing RNA secondary structures is hard. *Journal of Computational Biology*, 27(3):302–316, 2020. doi: 10.1089/cmb.2019.0420. prelim. RECOMB 2018; arXiv:1710.11513.

Anke Busch and Rolf Backofen. INFO-RNA—a fast approach to inverse RNA folding. *Bioinformatics*, 22(15):1823–1831, 2006. doi: 10.1093/bioinformatics/btl194.

Noam Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124, 1956. doi: 10.1109/TIT.1956.1056813.

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, and Pedro A. Ortega. Neural networks and the Chomsky hierarchy. In *ICLR*, 2023. arXiv:2207.02098.

Robin D. Dowell and Sean R. Eddy. Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. *BMC Bioinformatics*, 5:71, 2004. doi: 10.1186/1471-2105-5-71.

Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis*. Cambridge University Press, 1998.

Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J. R. Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, David Silver, Demis Hassabis, and Pushmeet Kohli. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53, 2022. doi: 10.1038/s41586-022-05172-4.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *ICML*, PMLR 202, pp. 10835–10866, 2023. arXiv:2210.10760.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, et al. Towards an AI co-scientist, 2025. arXiv:2502.18864.

Peter D. Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

Christine E. Hajdin, Stanislav Bellaousov, Wayne Huggins, Christopher W. Leonard, David H. Mathews, and Kevin M. Weeks. Accurate SHAPE-directed RNA secondary structure modeling, including pseudoknots. *PNAS*, 110(14):5498–5503, 2013. doi: 10.1073/pnas.1219988110.

John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.

Kexin Huang, Ying Jin, Ryan Li, Michael Y. Li, Emmanuel Candès, and Jure Leskovec. Automated hypothesis validation with agentic sequential falsifications. *arXiv preprint arXiv:2502.09858*, 2025. ICML 2025.

Stefan Janssen and Robert Giegerich. The RNA shapes studio. *Bioinformatics*, 31(3):423–425, 2015. doi: 10.1093/bioinformatics/btu649.

Yuki Kato, Hiroyuki Seki, and Tadao Kasami. RNA pseudoknotted structure prediction using stochastic multiple context-free grammar. *IPSJ Digital Courier*, 2:655–664, 2006. doi: 10.2197/ipsjdc.2.655.

Bjarne Knudsen and Jotun Hein. RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. *Bioinformatics*, 15(6):446–454, 1999. doi: 10.1093/bioinformatics/15.6.446.

Rohan V. Koodli, Boris Rudolfs, Hannah K. Wayment-Steele, and Rhiju Das. Redesigning the Eterna100 for the Vienna 2 folding engine. *bioRxiv*, 2021. doi: 10.1101/2021.08.26.457839. preprint.

Roham Koohestani, Ziyou Li, Anton Podkopaev, and Maliheh Izadi. Are agents just automata? on the formal equivalence between agentic AI and the Chomsky hierarchy. *arXiv preprint arXiv:2510.23487*, 2025.

Imre Lakatos. *The Methodology of Scientific Research Programmes*. Cambridge University Press, 1978.

Zijun Liu, Kaiming Liu, Yiqi Zhu, Xuanyu Lei, Zonghan Yang, Zhenhe Zhang, Peng Li, and Yang Liu. AIGS: Generating science from AI-powered automated falsification. *arXiv preprint arXiv:2411.11910*, 2024.

Ronny Lorenz, Stephan H. Bernhart, Christian Höner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F. Stadler, and Ivo L. Hofacker. ViennaRNA package 2.0. *Algorithms for Molecular Biology*, 6(1):26, 2011. doi: 10.1186/1748-7188-6-26.

Rune B. Lyngsø and Christian N. S. Pedersen. RNA pseudoknot prediction in energy-based models. *Journal of Computational Biology*, 7(3-4):409–427, 2000. doi: 10.1089/106652700750050862.

Daniel J. Mankowitz, Andrea Michi, Anton Zhernov, Marco Gelmi, Marco Selvi, Cosmin Paduraru, et al. Faster sorting algorithms discovered using deep reinforcement learning. *Nature*, 618(7964): 257–263, 2023. doi: 10.1038/s41586-023-06004-9.

John S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29(6-7):1105–1119, 1990. doi: 10.1002/bip.360290621.

Markus E. Nebel and Frank Weinberg. Algebraic and combinatorial properties of common RNA pseudoknot classes. *Journal of Computational Biology*, 19(10):1134–1150, 2012. PMC3469209.

Ruth Nussinov and Ann B. Jacobson. Fast algorithm for predicting the secondary structure of single-stranded RNA. *PNAS*, 77(11):6309–6313, 1980. doi: 10.1073/pnas.77.11.6309.

William Ogden. A helpful result for proving inherent ambiguity. *Mathematical Systems Theory*, 2 (3):191–194, 1968. doi: 10.1007/BF01694004.

Karl R. Popper. *The Logic of Scientific Discovery*. Hutchinson, London, 1959. transl. of Logik der Forschung, 1934.

Jens Reeder and Robert Giegerich. Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics. *BMC Bioinformatics*, 5:104, 2004. doi: 10.1186/1471-2105-5-104.

Elena Rivas and Sean R. Eddy. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *Journal of Molecular Biology*, 285(5):2053–2068, 1999. doi: 10.1006/jmbi.1998.2436.

Elena Rivas and Sean R. Eddy. The language of RNA: A formal grammar that includes pseudoknots. *Bioinformatics*, 16(4):334–340, 2000. doi: 10.1093/bioinformatics/16.4.334.

RNAInvBench. RNAInvBench and Pseudobase++: Pseudoknotted RNA inverse-design targets, 2024. inverse-design benchmark suite.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024. doi: 10.1038/s41586-023-06924-6.

Frederic Runge, Danny Stoll, Stefan Falkner, and Frank Hutter. Learning to design RNA. In *ICLR*, 2019. arXiv:1812.11951.

Yasubumi Sakakibara, Michael Brown, Richard Hughey, I. Saira Mian, Kimmen Sjölander, Rebecca C. Underwood, and David Haussler. Stochastic context-free grammars for tRNA modeling. *Nucleic Acids Research*, 22(23):5112–5120, 1994. doi: 10.1093/nar/22.23.5112.

Peter Schuster, Walter Fontana, Peter F. Stadler, and Ivo L. Hofacker. From sequences to shapes and back: A case study in RNA secondary structures. *Proceedings of the Royal Society of London B*, 255(1344):279–284, 1994. doi: 10.1098/rspb.1994.0040.

David B. Searls. The linguistics of DNA. *American Scientist*, 80(6):579–591, 1992.

David B. Searls. The language of genes. *Nature*, 420(6912):211–217, 2002. doi: 10.1038/nature01255.

Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. On multiple context-free grammars. *Theoretical Computer Science*, 88(2):191–229, 1991. doi: 10.1016/0304-3975(91)90374-B.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward hacking. In *NeurIPS*, volume 35, 2022. arXiv:2209.13085.

Fiona Y. Wang and Markus J. Buehler. Self-revising discovery systems for science: A categorical framework for agentic AI. *arXiv preprint arXiv:2606.01444*, 2026.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, et al. Voyager: An open-ended embodied agent with large language models, 2023. arXiv:2305.16291.

Joseph L. Watson, David Juergens, Nathaniel R. Bennett, et al. De novo design of protein structure and function with RFdiffusion. *Nature*, 620:1089–1100, 2023.

Jiaqi Wei, Yuejin Yang, Xiang Zhang, et al. From AI for science to agentic science: A survey on autonomous scientific discovery, 2025. arXiv:2508.14111.

Tomasz K. Wirecki, Grzegorz Lach, Nagendar Goud Badepally, S. Naeim Moafinejad, Farhang Jaryani, Gaja Klaudel, Kalina Nec, Eugene F. Baulin, and Janusz M. Bujnicki. DesiRNA: Structure-based design of RNA sequences with a replica exchange monte carlo approach. *Nucleic Acids Research*, 53(2), 2025. doi: 10.1093/nar/gkae1306.

Joseph N. Zadeh, Conrad D. Steenberg, Justin S. Bois, Brian R. Wolfe, Marshall B. Pierce, Asif R. Khan, Robert M. Dirks, and Niles A. Pierce. NUPACK: Analysis and design of nucleic acid systems. *Journal of Computational Chemistry*, 32(1):170–173, 2011. doi: 10.1002/jcc.21596.

Tianshuo Zhou, Ning Dai, Sizhen Li, Max Ward, David H. Mathews, and Liang Huang. RNA design via structure-aware multifrontier ensemble optimization. *Bioinformatics*, 39(Supplement 1):i563–i571, 2023. doi: 10.1093/bioinformatics/btad252.

Tianshuo Zhou, Wei Yu Tang, David H. Mathews, and Liang Huang. Undesignable RNA structure identification via rival structure generation and structure decomposition. In *RECOMB*, LNCS 14758, pp. 270–287, 2024. doi: 10.1007/978-1-0716-3989-4_17. arXiv:2311.08339.

Tianshuo Zhou, Apoorv Malik, Wei Yu Tang, David H. Mathews, and Liang Huang. Scalable and interpretable identification of minimal undesignable RNA structure motifs with rotational invariance. In *RECOMB*, 2025. arXiv:2402.17206.

Michael Zuker and Patrick Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9(1):133–148, 1981. doi: 10.1093/nar/9.1.133.

A Extended Technical Development and Experimental Record

This appendix retains the detailed definitions, experimental record, alternative estimators, and limitations of the full manuscript. The condensed main text states the principal claims; the material below documents their scope and provenance.

B Background and the Chomsky–RNA Correspondence

This section fixes the vocabulary the rest of the paper uses. Table 6 gathers the symbols and the non-standard terms in one place, so that nothing below has to be read in the order it was written; each is also introduced in prose where it is first needed.

Table 6: **Notation and terminology.** Symbols are grouped by what they describe; each is also defined at the point where it is first needed.

symbol / term	name	meaning
Structures and languages (§B, §D)		
Σ, x	alphabet, sequence	$\Sigma = \{A, C, G, U\}$; $x \in \Sigma^*$ is an RNA sequence
T, $P(T)$	target, pair set	a secondary structure and its set of base pairs
$\mathcal{T}$	structure space	all secondary structures over all lengths
$\mathcal{L}_{\text{nest}}$	nested language	structures with no crossing pair; context-free
$\mathcal{L}_{\text{pk}}$	crossing language	$\mathcal{T} \setminus \mathcal{L}_{\text{nest}}$, the structures with at least one crossing pair
$\mathcal{L}_{\text{H}}$	H-type family	the unbounded crossing family we construct; non-context-free, and generated by a dimension-2 MCFG
CF, MCF	class labels	context-free; mildly context-sensitive (multiple context-free)
Oracles, policies, and what they can reach (§B, §D)		
Ω	folding oracle	a map $\Sigma^* \to \mathcal{T}$ predicting the structure a sequence folds into
Ω_{cf}	CF oracle	the pseudoknot-free oracle (ViennaRNA MFE fold); $\operatorname{im}\Omega_{\text{cf}} \subseteq \mathcal{L}_{\text{nest}}$
Ω_{pk}	crossing-aware panel	the three-predictor panel {pKiss, ProbKnot, ShapeKnots}, abstaining on disagreement
π, π_{before}, π_{after}	design policy	a map $\mathcal{T} \to \Sigma^*$ proposing a sequence for a target; before/after the revision under audit
$\mathcal{R}_\Omega(\pi)$	realizable-target language	$\{T : \Omega(\pi(T)) = T\}$, the targets π realizes *as judged by* Ω; the object the negative side constrains
$\mathcal{O}$, $\Delta\mathcal{O}$	repertoire	the reusable components a policy searches over, and the ones the revision adds
$\mathcal{S}$	declared tool class	packaged inverse-design solvers, whose non-use is what "solver-free" asserts
A_{full}, A_{nest}	acceptance predicates	exact match of the *whole* pair set (used for every rate here) vs. of its nested projection only (reported only to count the errors it makes)
The audit's six components (§C, Eq. (4)–(5); verdicts in Table 7)		
$\mathcal{A}$	evidence vector	the audit's output: six findings with separate epistemic status, never one bit
E_{range}	exact exclusion	the theorem that the prior verifier's range contains no crossing structure
Δ_{prior}	gain over the prior policy	after minus before, under one shared adjudicator
Δ_{single}	gain over the single-oracle baseline	after minus the operator that optimizes one predictor, at matched oracle access
$\Delta_{\text{undirected}}$	gain over undirected search	after minus a budget-matched search with the objective deleted
$\Delta_{\text{held-out}}$	held-out-oracle gain	the same contrast scored by the one predictor held out of every loop

Table 6, continued.

symbol / term	name	meaning
g_{aut}	verified non-use	the $\pm$solver ablation over $\mathcal{S}$
Success rates, kept apart (§C, Eq. (6))		
r_{in}	in-objective rate	fraction of targets confirmed by pKiss $\wedge$ ProbKnot, both *inside* the panel operator's objective
r_{ho}	held-out rate	fraction confirmed by ShapeKnots alone, the predictor held out of every loop
r_{cons}	strict consensus rate	fraction confirmed by all three predictors; our primary, most stringent rate. It mixes two in-loop conjuncts with one held-out predictor, so it is *not* three-way independent confirmation
Δ_{bp}	search-free probe	the part of an effect that survives subtracting what random base-pair-compatible sequences already achieve
carry-over	held-out confirmation rate	P(the held-out predictor confirms | the arm's own in-loop predicate confirmed); the statistic the R1–R3 comparison uses
κ	Cohen's κ	chance-corrected agreement between two panel members
$\widehat{p}$	estimated rate	an estimated success rate, superscripted with the adjudicator and subscripted with the arm
$\mathcal{A}_{\text{match}} \supseteq \mathcal{A}_{\text{uMFE}} \supseteq \mathcal{A}_{\text{pfe}}$	verifier ladder	the nested "solved" predicates of the companion report (§B), from weakest to strongest: the target is the predicted structure; it is the *unique* minimum-free-energy structure; its Boltzmann probability exceeds a threshold
Arms and evidence rungs (§E; Table 8)		
single-oracle operator	the gamed arm	PKDECOMP optimizing pKiss alone; the arm whose $43/60$ collapses to $1/60$
panel operator	our hand-built operator	the same design loop re-optimized for pKiss $\wedge$ ProbKnot agreement
agent operator	LLM-written operator	one of the frozen operators written by a frontier model from primitives (App. J)
R1, R2, R3	evidence rungs	external adjudication; resource control; mechanistic attribution. A claim is placed on the highest rung its evidence reaches

Formal languages and the Chomsky hierarchy. A grammar generates a language of strings; the Chomsky hierarchy (Chomsky, 1956; Hopcroft & Ullman, 1979) orders grammars by generative power. Context-free grammars (CFGs) generate exactly the languages recognizable with one stack, of which balanced-bracket (Dyck) languages are canonical. The pumping and Ogden lemmas (Ogden, 1968; Hopcroft & Ullman, 1979) give necessary conditions for context-freeness, so a word family that cannot be pumped proves a language is not context-free. Multiple context-free grammars (MCFGs) (Seki et al., 1991) sit just above. They derive tuples of substrings at several positions, which is the machinery for crossing dependencies one stack cannot track, and dimension 2 suffices for the patterns we treat. The canonical witness is the count language $\{\, a^n b^m c^n d^m \,\}$, whose interleaved constraints force two substrings to grow in lockstep at separated positions (Hopcroft & Ullman, 1979). **We invoke this literature rather than extend it**, and use exactly two facts from it, both established elsewhere: the crossing family we target is non-context-free (Nebel & Weinberg, 2012), and specific pseudoknot classes sit in low-dimension MCFG subclasses (Kato et al., 2006). We make no claim that every pseudoknot topology is captured at dimension 2, and nothing below turns on the MCFG machinery beyond those two citations. Their role is to make the family of §E well defined, not to prove anything new.

RNA secondary structure, folding, inverse design. An RNA is $x \in \{A, C, G, U\}^n$; a *secondary structure* is a set P of base pairs (each index paired at most once). Pairs $(i, j), (k, l)$ with $i < k$ *cross* iff $i < k < j < l$; P is *nested* if none cross, *pseudoknotted* otherwise. A folding oracle maps

a sequence to a structure under a thermodynamic model, from base-pair maximization (Nussinov & Jacobson, 1980) through minimum-free-energy dynamic programming (Zuker & Stiegler, 1981) and the equilibrium partition function (McCaskill, 1990) to ViennaRNA (Lorenz et al., 2011). **We fix Ω_{cf} as a named function, not as a software package**: it is ViennaRNA's minimum-free-energy fold (`RNA.fold`, Turner parameters), whose dot-bracket output alphabet is $\{\,(\,,\,)\,,\,.\,\}$. This matters for the scope of Prop. 2 and we would rather state it here than have it read as an oversight. The same distribution ships `RNAPKplex`, which *does* emit crossings; it is never called anywhere on any path in this work, and if a prior system's verifier were `RNAPKplex` then $\operatorname{im}\Omega \not\subseteq \mathcal{L}_{\mathrm{nest}}$ and the exclusion would simply not apply to it. That is the correct behaviour of a scoped theorem, not a gap. The negative side is a statement about *a declared verifier*, and an audit that let the verifier be renegotiated after the fact would establish nothing. The *empirical* prior-policy floor is unaffected either way, since it is measured on the prior system as it actually ran. So Ω_{cf} is pseudoknot-free: its output space contains only nested structures ($\operatorname{im}\Omega_{\mathrm{cf}} \subseteq \mathcal{L}_{\mathrm{nest}}$), and it cannot represent a crossing pair. Inverse design (given T, find x with $\mathrm{fold}(x) = T$) is mature, with heuristic (Busch & Backofen, 2006), ensemble-defect (Zadeh et al., 2011), learned (Runge et al., 2019), ensemble-optimization (Zhou et al., 2023), and pseudoknot-capable (Wirecki et al., 2025) solvers. Pseudoknot-aware prediction oracles Ω_{pk}, pKiss (Janssen & Giegerich, 2015), ProbKnot (Bellaousov & Mathews, 2010), and ShapeKnots (Hajdin et al., 2013) in the lineage of (Rivas & Eddy, 1999), output crossings but are heuristic, not biophysical ground truth. Pseudoknot-free folding is polynomial; pseudoknotted folding is NP-complete (Lyngsø & Pedersen, 2000), and inverse design is NP-hard even in the minimal base-pair-maximization model (Bonnet et al., 2020). The sequence→structure map is many-to-one, with large neutral networks (Schuster et al., 1994; Aguirre et al., 2011); undesignability (no sequence realizes T) is provable by exhibiting, for every candidate, a rival structure at least as favorable (Zhou et al., 2024; 2025).

The verifier ladder (a reused instrument). A related technical report (Anonymous, 2026) turns "solved" into a nested family of decidable predicates on one target, $\mathcal{A}_{\mathrm{match}} \supseteq \mathcal{A}_{\mathrm{uMFE}} \supseteq \mathcal{A}_{\mathrm{pfe}}$, under Ω_{cf}, with a separate pseudoknot-aware branch for crossing targets. We reuse it as the adjudicating instrument and stake an orthogonal thesis (capability-class acquisition, not verifier over-acceptance). It also measured the pseudoknot verdict's oracle-dependence: the fraction of state-of-the-art pseudoknot designs judged "formed" swings across predictors (pKiss/ProbKnot/ShapeKnots) over a $[\,15, 60\,]\%$ envelope, which we carry into our positive-side design (§C.1).

B.1 RNA Topology Is Native to the Chomsky Hierarchy

Reading biological sequence structure as a formal language is a long tradition (Searls, 1992; 2002). Nested arcs are balanced brackets, context-free and generated by the RNA stochastic context-free grammar $S \to (\,S\,)\,S \mid .\,S \mid \varepsilon$ (Rivas & Eddy, 2000; Durbin et al., 1998), with SCFGs the standard nested-RNA model (Sakakibara et al., 1994; Knudsen & Hein, 1999; Dowell & Eddy, 2004) and thermodynamic pseudoknot folders operating over the mildly context-sensitive class (Reeder & Giegerich, 2004). Crossing arcs are provably not context-free (an Ogden-lemma argument on the interleaved dependencies rules out any CFG (Nebel & Weinberg, 2012)) and are captured by an MCFG of dimension ≤ 2; the canonical pseudoknot grammar of Rivas & Eddy (2000) is exactly such (Kato et al., 2006). The pseudoknot-free/pseudoknot split a design pipeline already treats as two verifier branches therefore straddles the CF → MCF boundary *for the families we treat*; we do not claim the whole crossing set $\mathcal{L}_{\mathrm{pk}}$ falls in one fixed low-dimension MCFG class.

Why one stack fails, concretely. Reading a strand $5' \to 3'$, the nested grammar $S \to (\,S\,)\,S \mid .\,S \mid \varepsilon$ folds with a single stack: each opening base is pushed and its partner pops it, and well-nesting guarantees the most-recently-opened pair closes first. A canonical H-type pseudoknot violates this. In dot-bracket form with two pair classes, ((((([[[[)))))]]]], the square-bracket stem opens while the round-bracket pairs are still unclosed and must be remembered past the closing of the round stem, requiring a second, independent stack. This is the copy-language pattern, not the Dyck pattern: an Ogden-lemma argument rules out every CFG (Nebel & Weinberg, 2012), whereas an MCFG carrying the two stems as a coordinated pair of substrings generates it at dimension 2 (Kato et al., 2006). The two verifier branches a design pipeline already maintains thus straddle a generative-class boundary between the nested family and the non-CF H-type family, developed formally in

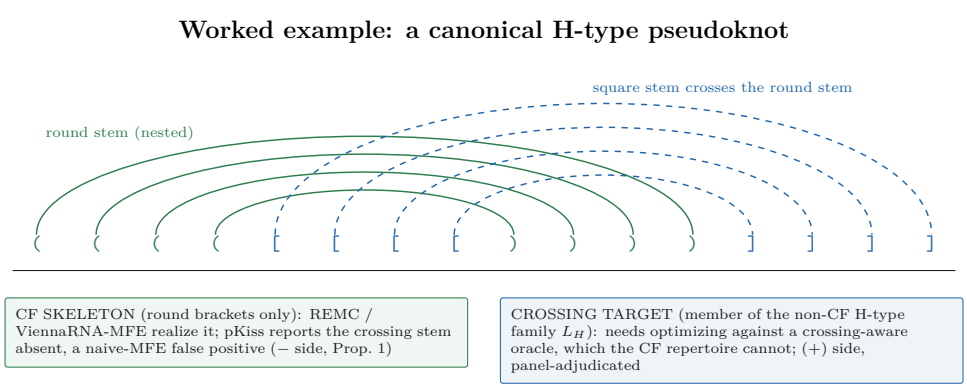

Figure 5: **One strand, two stems, and the exact point a pseudoknot-free verifier fails.** Arcs join paired bases on a canonical H-type target. The round stem nests, so a one-stack parser handles it and VIENNARNA/REMC realize it. The square stem opens *before* the round stem closes, so its partners must be remembered across a completed pop — the pattern one stack cannot track, and the reason no context-free grammar generates the family (Ogden lemma, Nebel & Weinberg, 2012) while a dimension-2 MCFG does (Kato et al., 2006). A pseudoknot-free oracle does not *fail to find* such a design: it reports the crossing stem absent and returns a naive-MFE false positive, which is the negative side of the audit in concrete form (Prop. 2). Everything the paper calls a "crossing target" is a member of this family or a larger relative of it.

§D. Figure 5 is that paragraph drawn: the same strand, the two stems, and the single place where a one-stack parser has to give up.

Generative power is the main line; folding complexity is a distinct axis. The Chomsky hierarchy measures generative power (which structures a grammar produces); computational complexity measures the cost of parsing and folding. Pseudoknotted folding is NP-complete and inverse design is NP-hard (Lyngsø & Pedersen, 2000; Bonnet et al., 2020). But that hardness is a *distinct* axis. It depends on the allowed pseudoknot topology, the energy model and the optimization objective, and it is not implied by the generative class: fixed fan-out MCFGs, for instance, parse in polynomial time. Our sole main-line quantity is the generative jump $\mathrm{CF} \to \mathrm{MCF}$; we treat the two axes as related but independent.

C THE TWO-SIDED CAPABILITY AUDIT

We now define the instrument. The setting is a design agent that, after some revision to itself, is claimed to realize targets its prior self could not. The audit asks two questions about that claim and keeps the answers apart: what the prior system's *verifier* was able to certify, which on this task is decidable, and what the revised system can actually do, which is not. This section states the six quantities that answer those questions (Def. 1), the rule that classifies each of them, and the two filled-in verdict tables that result — Table 7 for our own system and Table 8 for the agent-written arm.

One convention is fixed before anything else, because every rate below depends on it. *Targets are graded before anything is run.* Each target receives an intrinsic difficulty grade: its crossing-stem count, and whether it is nested or crossing. The grade is computed from the target's own pair set, never from whether the agent solved it, so the graded frontier carries no survivorship bias. Crossing targets then split by grade into the pools of Table 15.

Definition 1 (The audit's output: an evidence vector)**.** *Let π_{before} and π_{after} be the prior and the revised design policy, let Ω_{cf} be the prior system's pseudoknot-free verifier, and let Ω_{pk} be the crossing-aware panel. The audit of the claim "π_{after} realizes crossing targets that π_{before} could not" returns the six-component* evidence vector

$$\mathcal{A} = \big(E_{\mathrm{range}},\ \Delta_{\mathrm{prior}},\ \Delta_{\mathrm{single}},\ \Delta_{\mathrm{undirected}},\ \Delta_{\mathrm{held\text{-}out}},\ g_{\mathrm{aut}} \big), \tag{4}$$

whose components are defined in Eq. (5) *below. A verdict is a statement about which components are established and which are not, never a single bit.*

Why a vector and not a pass/fail bit. A pass/fail conjunction would be the wrong output. A single threshold on a positive rate is passed by one lucky target. It encodes none of what a reader needs: the separation from a gaming baseline, the separation from matched undirected search, the spread across targets, the uncertainty, or whether a predictor outside the loop agrees. The six components are also of *different epistemic kinds*: one is a theorem and the other five are effect sizes with intervals, so collapsing them would destroy the only information the audit adds. Underlying all five Δ's is one quantity, the held-out positive rate. On held-out crossing targets the system must produce designs that a crossing-aware predictor panel confirms fold to the target's *entire* pair set. That rate is reported with its uncertainty and its spread across targets, never against a threshold.

What "capability acquisition" names here. It names the class of claim this instrument grades, not a claim we make. By our own standard (Table 8) no arm in this paper reaches R3, so nothing here is asserted to be an acquired design principle; what is asserted is a measured difference together with the alternatives it excludes. Relatedly, the phrase CF $\rightarrow$ MCF names a difficulty axis over target *families*, never a proven change of Chomsky class in the policy's own realizable-target language. A positive finding is a per-target verified event, and any finite set of realized targets is regular. No finite experiment could establish the class-level statement (§D).

The six components. Writing $\widehat{p}^{\,\mathrm{panel}}$ for a panel-confirmed success rate and r_{ho} for the rate under the held-out predictor alone (Eq. (6)),

$$
\begin{aligned}
E_{\mathrm{range}} &= \left[\, \mathcal{R}_{\Omega_{\mathrm{cf}}}(\pi_{\mathrm{before}}) \cap \mathcal{L}_{\mathrm{pk}} = \varnothing \,\right] && \text{exact (Prop. 2), discharged before any run} \\
\Delta_{\mathrm{prior}} &= \widehat{p}^{\,\mathrm{panel}}_{\mathrm{after}} - \widehat{p}^{\,\mathrm{panel}}_{\mathrm{before}} && \text{before/after under one } \textit{shared} \text{ adjudicator} \\
\Delta_{\mathrm{single}} &= \widehat{p}^{\,\mathrm{panel}}_{\mathrm{after}} - \widehat{p}^{\,\mathrm{panel}}_{\mathrm{single}} && \text{vs. single-oracle gaming, matched access} \\
\Delta_{\mathrm{undirected}} &= \widehat{p}^{\,\mathrm{panel}}_{\mathrm{after}} - \widehat{p}^{\,\mathrm{panel}}_{\mathrm{undir}} && \text{vs. matched-nominal-cap undirected search} \\
\Delta_{\text{held-out}} &= r^{\mathrm{after}}_{\mathrm{ho}} - r^{\mathrm{single}}_{\mathrm{ho}} && \text{on the one genuinely held-out predictor} \\
g_{\mathrm{aut}} &= \widehat{p}_{\mathrm{after}} - \widehat{p}^{\,-\mathrm{solver}}_{\mathrm{after}} && \text{verified non-use of the solver class } \mathcal{S}.
\end{aligned}
\tag{5}
$$

Only E_{range} is decidable and discharged before any run; the five Δ's are measured, model-relative, and each is reported with an effect size and an interval. **A study may establish some components and fail to establish others, as ours does.** The audit's job is to say which, not to collapse them. Table 7 is this paper's own filled-in $\mathcal{A}$, stated up front so no reader has to reconstruct it from the results section.

How to read a mostly-negative vector. It is tempting to read Table 7 as a report card our system failed. The system audited here was *built to pass this audit*: it optimizes a two-oracle conjunction because a single oracle is gameable. It still leaves two of six components unestablished or weak. But our own failure to saturate the instrument says nothing about the instrument until some arm succeeds. The discriminating power is established by the positive control of §E, not by our low score.

What each side controls for, and what none of them does. The limit first, since it is the most easily overclaimed. *This audit does not separate capability from search.* The operator we audit *is* a stochastic search against folding oracles, and our own ablation attributes its gain to oracle-guided repair (Table 12). Nor should the two be opposed. A finite-description search procedure that reliably realizes held-out family members is a perfectly good candidate capability. What the audit controls for is narrower, and each component excludes a named alternative: target memorization and one-off luck (only partly, since our split is disjoint by target identity and not by structure), matched-cap undirected search, single-oracle gaming, and delegation to a packaged solver. No single component supports the word *acquisition*, and neither does their conjunction when some are unestablished, so the output is the vector of Eq. (4) read component-wise. The structure parallels the retrieval/search/discovery distinction of Wang & Buehler (2026), with one amendment. We place our result at *search*, and the audit's job is to *ask* whether that search is transferable, separated from

undirected search and undelegated, not to certify that it is. On this study only the last comes back positive, and only as verified non-use.

Terminology. **Audit** names the whole protocol of Def. 1, never a single result. Within it, **exact exclusion** is the verifier-range theorem (Prop. 2), the only exact component; **empirical gain** is the before/after difference under a shared adjudicator; **verified non-use** is the solver-removal result, relative to a declared tool class and an audit of the call graph rather than a counterfactual (Rem. 1); **cross-predictor consistency** is what the panel measures, never physical truth; and **scientific confirmation**, the wet-lab rung, is not attempted. **We reserve "certified" for the exact exclusion alone**: a panel verdict is *adjudicated* or *mixed-panel-consistent*, and no design here is called certified. For the same reason we write *mixed-panel-consistent crossing* and not "panel-robust ascent", since two of three predictors are inside the operator's objective and the third has $\sim 1\%$ native recall.

Three metrics, named and kept apart. A single "panel" number conflates quantities of different evidential value, so we name all three and report them separately throughout:

$$\begin{aligned} r_{\text{in}} &: \text{pKiss} \wedge \text{ProbKnot} \;\; \text{(both } \textit{in} \text{ the objective)} && \Rightarrow \text{ in-objective consistency} \\ r_{\text{ho}} &: \text{ShapeKnots alone} \;\; \text{(the one oracle held out)} && \Rightarrow \text{ held-out-oracle generalization} \\ r_{\text{cons}} &: \text{all three} \;\; \text{(two in-loop, one held out)} && \Rightarrow \text{ mixed in-loop/held-out.} \end{aligned} \tag{6}$$

For the panel operator, r_{in} scores its own objective and so cannot evidence generalization; r_{ho} is the only genuine held-out-oracle rate and is *weak* (§E); r_{cons}, the rate we report as primary because it is the most stringent, is a *mixture* of two in-loop conjuncts plus one held-out, and must not be read as three-way independent confirmation. Every headline number below is labelled with which of the three it is. Relatedly, **"held-out" in this paper means held out of the optimization *loop*** (the ShapeKnots adjudicator), and never "held out of development". The primary $n{=}60$ pool is a *development/evaluation* pool: the operator and every hyperparameter were fixed on it. Pools are named in Table 15 and we use those names.

Classification rule, and how to count the components. *Exact*: proved, discharged offline. *Established*: the effect's 95% interval excludes zero at the deepest resampling level the component's design contains. *Weak*: directionally consistent, but the interval or the replication does not support it. *Not established*: the interval spans zero. By that rule this study returns **one exact, three established, one weak, one not established**, and every count stated elsewhere in this paper is this one.

Two caveats attach to applying that rule. Where a component admits more than one nesting variant, we name which is primary before seeing the answer. For Δ_{single} the primary estimator draws all three executions inside each drawn seed, because that is the design that was run; the one-execution-per-seed variant is a conservative sensitivity. We hold no timestamped pre-registration, so "fixed in advance" is our word, and the choice between variants is a degree of freedom a reader cannot audit. The resolution we would defend is not the choice but the deterministic re-run that removes it, which has no execution level to nest (Table 12). And the count should not be read without the second question Table 7 answers, flagged NO COUNTERFACTUAL and WEAK COUNTERFACTUAL: whether a component contrasts against an arm that could have come out otherwise. Two do not. E_{range} is a theorem and g_{aut} records that a solver was never on the method path, so neither can be surprised by data; Δ_{prior} is weak in this sense because the prior arm optimizes a different objective. Only Δ_{single}, $\Delta_{\text{undirected}}$ and $\Delta_{\text{held-out}}$ are counterfactual contrasts, and of those one is narrow, one null and one weak. The rungs of Table 8 use the same words for a different arm and are not part of this count.

Table 7: **The audit's output for this study: the evidence vector $\mathcal{A}$ of Eq. (4), filled in.** All rates are r_{cons} (Eq. (6)) unless stated. This is the verdict on *our own system*: *partial audit success: diagnostic crossing evidence, not robust capability acquisition*. Table 8 reports the arm that does clear the bar, which this vector has no row for. Each row is a pointer, not an argument: the classification rule is in §C and every number is derived where it is cited.

component	verdict	basis	what it does *not* show
E_{range} exact exclusion	**exact**	Prop. 2, discharged offline before any run	Bounds the prior *verifier's* range, not the prior policy's capacity. NO COUNTERFACTUAL: a theorem cannot be surprised by data
Δ_{prior} prior floor	**established**	Prior policy $0/60$ at *every* one of 24 executions; after, 1–9/60 (mean 3.29) under the shared panel	The quoted $+0.017$–$+0.15$ is the **range of point estimates**, not a 95% interval. So this row rests on the floor being an exact zero, not on the interval rule. WEAK COUNTERFACTUAL: the prior arm optimizes a different objective
Δ_{single} vs. single-oracle gaming	**narrow**	*Established, narrowly.* $+0.067$ $[+0.023, +0.120]$ bit-reproducibly, single-oracle $0/60$ at all five seeds (Table 12); operator strictly higher in $14/15$ wall-clock executions and $15/15$ on the structurally disjoint pool (§E)	A stable *magnitude*. The estimate fell $+0.080 \to +0.056 \to +0.048$ as replication was added, and determinism removes the execution level, not the target set. A search-free probe puts 84% of it where a random compatible sequence already succeeds, leaving $+0.016$. Between-system; matched on oracle *access*, not *use* ($2.3\times$)
$\Delta_{\text{undirected}}$ vs. matched undirected search	**not est.**	*Not established.* Exactly $+0.0000$ on 300 paired bit-reproducible cells, $[-0.060, +0.050]$ (Table 12)	An underpowered null. Execution noise can no longer be blamed, so this is *measured absent* at this pool size. Deleting the objective does not lower r_{cons} at all; only deleting the repair loop does. Matched on allowance, not on realized use
$\Delta_{\text{held-out}}$ held-out oracle	**weak**	r_{ho} higher at every seed but significant in 1 of 3 (§E)	A small *absolute* signal: against its own ceiling the judge confirms our designs at 8.0% where it recovers only 11.3% of natives, against the single-oracle operator's 3.7% (App. H.2). Run in an unvalidated de-novo mode; the comparison is unstratified
g_{aut} verified non-use of packaged solvers	**established**	No packaged inverse-design solver is ever called	Anything counterfactual: the solver was never on the method path, so removing it changes nothing. Solver-free is *not* oracle-free, and not evidence that the gain is the system's own

Table 8: **The instrument's other output, which the vector above cannot carry.** Table 7 is the audit's verdict on *our* system, so by construction it has no row for the arm that does clear the bar. That arm is the frozen LLM-written operators of App. J, which optimize the same pKiss $\wedge$ ProbKnot predicate and are scored by the ShapeKnots predictor held out of every arm's loop. We report them on the three independently reachable rungs of §6. It is simultaneously the positive control that shows the panel is not merely insensitive and the sharpest bound on our own result. Quantifier: these are the 2 of 6 invented operators with no runtime failures, so the claim is about *some* agent-written operators, not the class.

rung	verdict	basis	what it does *not* show
(R1) EXTERNAL ADJUDICATION the difference shows up under a judge no arm's objective can flatter	**established**	Paired and difficulty-matched on the 951 units *both* arms solved under the shared predicate: 0.293 vs. 0.095, difference $+0.199$, target-clustered $[+0.108, +0.297]$ over 42 clusters, cluster-permutation $p{=}5\times10^{-5}$. Re-derived per design through a second code path with zero disagreements	Correctness, and not a within-system contrast. Carry-over measures agreement with a partly correlated panel (§C.1). We quote the clustered pair, not the exact McNemar $p{=}5\times10^{-37}$ the same counts give, since the 951 units come from 60 targets. One pool; the model-level selection above
(R2) RESOURCE CONTROL the difference is not bought with compute	**established**	Measured in both directions. Tripling the pKiss allowance ($82.2\rightarrow196.2$ calls/target) moves carry-over $0.108\rightarrow0.108$. And the agents' *realized* use is ≈28 and ≈13 calls/target against this operator's 128.0; an arm held to that volume (26.0) carries over at 0.107 where the agent reaches 0.367	That no resource explains it. Only that oracle-call volume does not, in either direction. Resources we did not count remain open
(R3) MECHANISTIC ATTRIBUTION the mechanism is identified and transfers	**not est.**	*Not established, informatively.* **Seven** mechanisms transplanted one at a time, deterministically at 5 seeds: four legible in the source, a compute control in each direction, and *two* properties that differ between the arms at matched composition and predict or explain carry-over observationally. Every interval contains the baseline; none reaches the agents' lower bound of 0.275. The two observational leads are the informative part: each would have been reported as the mechanism by an analysis that stopped before intervening (§E)	What the difference *is*. We withhold the word "discovery" until this rung is reached, and note in §5 that the criterion we position against reaches neither R1 nor R2

The audit is asymmetric. The negative side has two components of different epistemic status: an *exact*, decidable verifier-range bound (the representational limit of Ω_{cf}, Prop. 2) and a *separate*, empirical prior-policy floor ($0/60$ under the shared crossing-aware panel). Only the first is a theorem. The positive side is also not exact: Ω_{pk} is a heuristic, model-relative predictor whose verdict is oracle-dependent (§B,C.1). We therefore claim a decidable negative side and a model-relative, panel-adjudicated positive side, not a uniformly "exact" audit, and report the positive rate with its cross-oracle envelope (§C.1).

Table 9: r_{cons} **separated from single-oracle gaming**, matched design (n=60 development/evaluation pool; panel-unanimous = pKiss ∧ ProbKnot ∧ ShapeKnots, all exact). The single-oracle and panel operators run on the *same* five seeds and targets. b solved only by the panel operator, c only by the single-oracle operator; $\widehat{\Delta}_s$ is the per-seed paired risk difference. Note c=0 at every seed. Single 60-target seeds are underpowered, and this $+0.080$ is the *first* of four estimates of Δ_{single}; §E gives the sequence and what it does and does not settle. (`stats_separation.py`, `large_pool_stats.py`.)

seed	**single r_{cons}/60**	**panel r_{cons}/60**	**paired McNemar** (b, c, p)	$\widehat{\Delta}_s$
seed 0	1	8	b=7, c=0, p=0.016	+0.117
seed 1	0	3	b=3, c=0, p=0.25	+0.050
seed 2	0	5	b=5, c=0, p=0.0625	+0.083
seed 3	0	5	b=5, c=0, p=0.0625	+0.083
seed 4	0	4	b=4, c=0, p=0.125	+0.067
pooled	1/300	25/300	Δ=+.080; CI [+.040, +.127] / two-way [+.030, +.140]	

Sealed-commitment protocol. We distinguish three oracle roles: the *optimization* oracles the operator queries inside its repair loop (pKiss, and ProbKnot for the panel operator), the *selection* of a final candidate, and the *held-out adjudicator* (chiefly ShapeKnots), held out of every loop. For each target the agent commits a hash-frozen prediction, either a candidate sequence or an explicit abstention claim (§D.1), before the held-out adjudicator is run, so the prediction is risky against an oracle it never optimized and post-hoc selection there is foreclosed. Sealing does *not* prevent overfitting to the in-loop oracles; our $43/60 \to 1/60$ collapse is exactly such overfitting. *So what does sealing buy if the dominant failure is one it does not prevent?* The ability to see that failure. Without a seal an arm may select among candidates using the held-out predictor, and any collapse it would have shown is selected away before adjudication, making $43 \to 1$ unobservable in principle rather than merely unobserved. Sealing does not stop in-loop overfitting. It makes in-loop overfitting *measurable*. This operationalizes a Lakatosian risky prediction (Popper, 1959; Lakatos, 1978), the design-time analogue of pre-registration and of agentic falsification loops (Huang et al., 2025; Liu et al., 2024), but with an exact refuter, not a statistical one.

The certification procedure and its soundness. Auditing an ascent requires each component to be computed by a stated procedure, not judged. E_{range} is discharged offline by Prop. 2: no search, no budget, no seed. Each Δ is a paired difference between two arms scored by the *same* adjudicator on the *same* targets inside the *same* execution, so common-mode noise cancels; the pairing, resampling unit and interval type are fixed in App. G. g_{aut} is a $\pm$solver ablation over the declared tool class. Soundness is conditional on the panel: a panel that errs systematically moves every positive component with it, so it abstains on disagreement, why one predictor is kept out of every loop, and why §E reports a positive control. No procedure here converts a model-relative verdict into a physical one.

The verdict is a demarcation, not a probability, and it is component-wise: E_{range} is a theorem (Prop. 2), whereas the five Δ's are model-relative measurements whose uncertainty we quantify (§E, Tables 9, 12) and two of which this study does not establish (Table 7).

C.1 The Positive Side Is Model-Relative

Only the negative side's verifier-range bound is exact (Prop. 2); its prior-policy floor is empirical, and the positive side is neither exact nor decidable. A "formed pseudoknot" verdict comes from a heuristic predictor over a restricted pseudoknot class, not from biophysical ground truth, and it is oracle-dependent. A related technical report measured the fraction of state-of-the-art pseudoknot designs judged to form their target swinging across a $[15, 60]\%$ envelope over pKiss, ProbKnot and ShapeKnots, with only fair pairwise agreement (Anonymous, 2026). That disagreement is mechanistic and not noise, since the three cover different pseudoknot subclasses and optimize different objectives. pKiss enumerates H-type and kissing-hairpin motifs under a heuristic folding model (Janssen & Giegerich, 2015), ProbKnot assembles maximum-expected-accuracy pairs from a par-

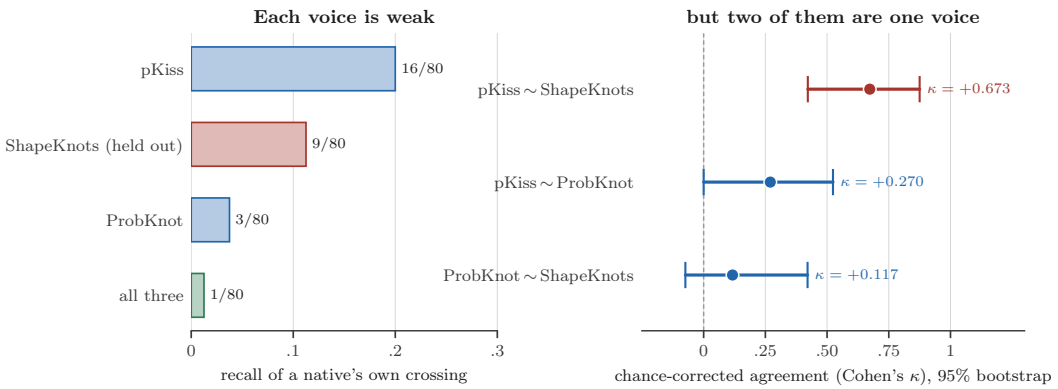

Figure 6: **The panel is structured, not exchangeable** (measured on the 80 native bpRNA pseudoknots of the calibration set; `panel_agreement.py`). *Left:* each predictor folding a native's *own* sequence recovers its known crossing rarely — $16/80$, $9/80$, $3/80$ — and all three agree on only $1/80$. These marginals are why we score agreement with Cohen's κ: two predictors that mostly say "no" agree $\sim$84% of the time by construction. *Right:* chance-corrected pairwise agreement with 95% bootstrap intervals. pKiss and ShapeKnots form a correlated block whose interval clears zero, and the two pairings involving ProbKnot do not. Since ShapeKnots is the held-out predictor and pKiss is what the single-oracle arm optimizes, the held-out verdict is the *least* independent one available — a bound on the *absolute* reading of carry-over, though not on the paired R1 comparison, where both arms meet the same in-loop bar and the shared correlation cancels. ProbKnot's $3/80$ leaves every κ involving it poorly determined, so we do not claim "two voices, not three"; we claim the panel is worth less than its arity, and owed experiment (3) asks for a predictor outside this model family.

tition function (Bellaousov & Mathews, 2010), and ShapeKnots folds under experimental probing constraints (Hajdin et al., 2013). A single-predictor rate is therefore class-biased, and a rate under one lenient predictor would be an oracle artifact. Abstaining when the panel disagrees declines to convert a modeling artifact into a claimed capability. We call the predictors *distinct* and not statistically independent, since all three are computational secondary-structure models sharing thermodynamic and structural assumptions, so their errors can correlate. Panel-unanimity is cross-predictor consistency under the declared computational panel, not independent physical confirmation.

How many independent voices does the panel have? (Measured, not argued.) We measure pairwise agreement on the 80 native bpRNA pseudoknots of our calibration set, where each predictor's call is whether folding a native's *own* sequence recovers its known crossing (`panel_agreement.py`). We use Cohen's κ, not raw agreement, because at these marginal rates (pKiss 20%, ProbKnot 4%, ShapeKnots 11%) two predictors that mostly say "no" agree $\sim 84\%$ of the time by construction. The panel is **not three exchangeable voices**: pKiss and ShapeKnots agree at $\kappa = 0.673$ $[0.422, 0.875]$, an interval clear of zero, so they behave as a correlated block, while the two pairings involving ProbKnot are $\kappa = 0.270$ $[0.000, 0.524]$ and $\kappa = 0.117$ $[-0.074, 0.421]$. We resist reading this as "two voices, not three": ProbKnot recovers only 3 of 80 natives, so every κ involving it is poorly determined and whether it is a genuinely independent third voice cannot be settled at n=80. The defensible statement is weaker and still consequential: at least two of three members are substantially correlated, so unanimity buys less independent confirmation than its arity suggests. Figure 6 shows both halves of that measurement at once, and the uncomfortable part is the right-hand one: the predictor we hold out is the one most correlated with the predictor the arms optimize.

This also gives the low r_{cons} a mechanism instead of leaving it a bare number, since the three-way conjunction is bound largely by ProbKnot, which is exactly where the single-oracle operator's designs die ($2/43$, Table 11).

Does that correlation undercut R1? Not the comparison, and the reason is structural. Both arms in the R1 comparison optimize the *same* predicate and the comparison is conditioned on both having satisfied it on the same target, so whatever agreement the correlation hands out is handed to both

arms equally and cancels in the paired difference. What it does bound is the *absolute* reading: carry-over measures agreement with a partly correlated panel, not correctness, so 0.293 is not an estimate of how often those designs really fold as intended. The same distinction disposes of the mirror worry about Δ_{single}, where the two arms optimize *different* predicates so the correlation does not cancel, which is one more reason we treat that component as narrow and this one as established.

Then why not make the least-correlated pair the primary metric? The answer is structural, not a preference. ProbKnot is the member least correlated with the others, so pKiss $\wedge$ ProbKnot is indeed the most mechanistically diverse pairing available. But that conjunction *is* r_{in}, the panel operator's own success predicate, so for that arm it cannot serve as a held-out measure at all; scoring an arm on its own objective is the failure this whole paper is about. Where ProbKnot *is* held out we already use it as primary, and it carries the paper's sharpest single result: the single-oracle operator optimizes pKiss alone, so its $2/43$ against the minimum-free-energy solver's $26/43$ is measured on exactly that least-correlated predictor (Table 11). The pairing the question asks for is therefore used wherever it is available and unavailable for the one arm whose objective it is. What no rearrangement of three thermodynamic predictors fixes is the underlying problem, which is why owed experiment (3) asks for a predictor from a different model family rather than a different conjunction of these.

We adjudicate the positive side accordingly. A held-out crossing target counts as realized only under an explicit rule, and we report all three: (i) *panel-unanimous* (all three predictors confirm the exact crossing pair set) as the primary rate; (ii) the reference (pKiss) rate; and (iii) the cross-predictor envelope $[\min, \max]$. Targets on which predictors disagree are abstained rather than counted either way. The audit is thus asymmetric by construction: a negative side exact and decidable on its verifier-range component (plus an empirical floor) and a model-relative, panel-adjudicated, abstaining positive side.

D Formalization: the Realizable-Target Language

A pumping lemma constrains languages of structures; an agent manipulates sequences. The object the lemma acts on is therefore not the agent's operators but the set of target structures the agent realizes under an oracle. This fixes a category error in a naive "operators place pairs" account: an operator edits a sequence, and whether the result is nested or crossing is a property of the fold, decided by the oracle, not of the operator.

Policies, oracles, and the realizable-target language.

Definition 2 (Structures, oracles, policies)**.** *Fix* $\Sigma = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}$ *and let* $\mathcal{T} = \bigcup_n \mathcal{T}_n$ *be the dot-bracket structures, partitioned into the* nested *language* $\mathcal{L}_{\text{nest}}$ *(no two base pairs cross) and the* crossing *language* $\mathcal{L}_{\text{pk}} = \mathcal{T} \setminus \mathcal{L}_{\text{nest}}$*. A* folding oracle *is a length-preserving map* $\Omega : \Sigma^* \to \mathcal{T}$*; a* design policy *is a length-preserving map* $\pi : \mathcal{T} \to \Sigma^*$*, realized by a search over a repertoire of reusable components* $\mathcal{O}$*. The* realizable-target language *of* π *under* Ω *and the* designable set *of* Ω *are*

$$\mathcal{R}_\Omega(\pi) = \{\, T \in \mathcal{T} : \Omega(\pi(T)) = T \,\}, \qquad \mathcal{D}_\Omega = \operatorname{im}\Omega = \{\, T : \exists\, x \in \Sigma^*,\ \Omega(x) = T \,\}. \tag{7}$$

$\mathcal{R}_\Omega(\pi)$ *is a language over dot-bracket strings, the object a Chomsky-class statement must range over, since a pumping lemma constrains languages of* structures*, not the sequence edits an operator performs.*

Lemma 1 (Realizability is capped by the oracle range)**.** *For every policy* π *and oracle* Ω*,* $\mathcal{R}_\Omega(\pi) \subseteq \mathcal{D}_\Omega = \operatorname{im}\Omega$*, independently of the policy's search budget or repertoire.*

Proof. If $T \in \mathcal{R}_\Omega(\pi)$ then $T = \Omega(\pi(T))$, so T is in the image of Ω; i.e. $T \in \operatorname{im}\Omega = \mathcal{D}_\Omega$. The bound quantifies over *all* π, so no amount of search enlarges $\mathcal{R}_\Omega(\pi)$ beyond $\operatorname{im}\Omega$. □

Lemma 1 is the engine of the negative side: it makes confinement a property of the oracle's representation, not of any operator, and reduces Prop. 2 to computing $\operatorname{im}\Omega_{\text{cf}}$.

Where this object sits in designability theory. $\mathcal{R}_\Omega(\pi)$ is bounded above by the designable structures under Ω (those admitting some sequence that folds to them), and inherits the non-uniformity of the sequence-to-structure map: designable structures form vast neutral networks, while many

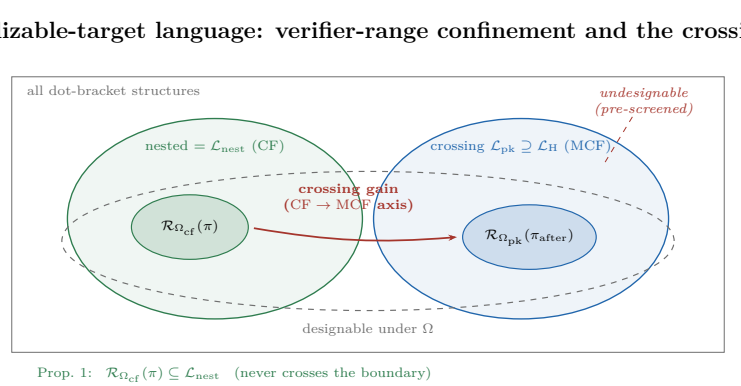

Figure 7: **The realizable-target language, confinement, and the ascent.** Nested structures form a context-free region $\mathcal{L}_{\text{nest}}$ and crossing structures a disjoint region $\mathcal{L}_{\text{pk}}$ (whose H-type sub-family $\mathcal{L}_{\text{H}}$ is non-CF, strictly higher); the designable set (dashed) is the subset realized by some sequence. Under the pseudoknot-free oracle, the realizable-target language $\mathcal{R}_{\Omega_{\text{cf}}}(\pi)$ is confined to $\mathcal{L}_{\text{nest}}$ (Prop. 2) and cannot enter $\mathcal{L}_{\text{pk}}$; an empirically observed crossing-family gain moves the measured $\mathcal{R}_{\Omega_{\text{pk}}}(\pi_{\text{after}})$ into the designable part of $\mathcal{L}_{\text{pk}}$, while undesignable crossing targets are pre-screened out.

structures are realized by no sequence (Schuster et al., 1994; Aguirre et al., 2011). Its extreme is undesignability, where no sequence realizes T under the energy criterion, provable by construction (Zhou et al., 2024) and the complement of the designable set. Hence a protocol consequence (§E): a held-out crossing pool must be pre-screened for designability, so an intrinsically unrealizable target is scored as an impossibility (§D.1), not a capability failure. The realizable-target language is thus the intersection of what is designable in principle and what π actually finds; an observed crossing gain concerns the second factor, over targets the first admits (Fig. 7).

Proposition 2 (Decidable CF-confinement)**.** *The pseudoknot-free oracle satisfies* $\operatorname{im}\Omega_{\text{cf}} \subseteq \mathcal{L}_{\text{nest}}$ *(every single-structure output it returns is nested, admitting no crossing pair; the image is in fact the strictly smaller* designable-*nested subset, but* $\subseteq$ *is all the negative side needs). Hence, by Lemma 1,* $\mathcal{R}_{\Omega_{\text{cf}}}(\pi) \subseteq \mathcal{L}_{\text{nest}}$ *for* every *policy* π*, and for every crossing target* $T \in \mathcal{L}_{\text{pk}}$ *we have* $T \notin \mathcal{R}_{\Omega_{\text{cf}}}(\pi)$*. This non-membership is* decidable *in* $O(|P|^2)$ *(test whether any two pairs of* T *cross; App. F) and is independent of the policy's search budget.*

The proof (App. F) is immediate from the oracle's output space: the negative side of Def. 1 is a theorem discharged before any run, not a failed search.

What the Chomsky framing buys, and what it does not. It buys three things: an *externally fixed* difficulty axis that we did not choose to flatter the result, a decidable negative side, and an unbounded family over which extrapolation is meaningful. It does *not* buy four others, and we name them because each has been read into work of this kind. (i) It does not show the agent's policy changed Chomsky class: any finite set of realized designs is regular, so no finite experiment can. (ii) It does not make the positive side exact, since crossing realization is adjudicated by heuristic predictors, and only the exclusion is a theorem. (iii) It does not license "the agent learned a grammar": we measure realized targets, not induced rules. (iv) It does not make the crossing-pair count a grammar dimension; the count is a structural covariate that happens to track difficulty. The framing is a source of exactness on one side and a vocabulary on the other, and we keep those roles apart throughout.

The boundary is a genuine class boundary (invoked for the general results, verified only for our own family). $\mathcal{L}_{\text{nest}}$ is context-free (Rivas & Eddy, 2000; Durbin et al., 1998). We use $\mathcal{L}_{\text{pk}} = \mathcal{T} \setminus \mathcal{L}_{\text{nest}}$ for *all* crossing structures (a structural set, used as such on the negative side). The formal-language claim is scoped to the specific unbounded family: the H-type family $\mathcal{L}_{\text{H}} \subseteq \mathcal{L}_{\text{pk}}$ is non-context-free (Nebel & Weinberg, 2012) and MCFG of dimension 2 (Rivas & Eddy, 2000; Kato et al., 2006); we do *not* claim the whole crossing set $\mathcal{L}_{\text{pk}}$ is uniformly MCFG-2 (richer topologies may need higher dimension, §B). We take these as established and tie the agent to them via Prop. 2:

"could not before" is $\mathcal{R}_{\Omega_{\mathrm{cf}}}(\pi_{\mathrm{before}}) \subseteq \mathcal{L}_{\mathrm{nest}} \not\ni T$; "can now" is $T \in \mathcal{R}_{\Omega_{\mathrm{pk}}}(\pi_{\mathrm{after}})$ for a crossing $T \in \mathcal{L}_{\mathrm{pk}}$ (the class-level claim, §D, concerns the non-CF family $\mathcal{L}_{\mathrm{H}}$).

Definition 3 (Empirical crossing-family capability gain (fixed oracle))**.** *Let $\mathcal{D}_{\mathrm{ho}} \subseteq \mathcal{L}_{\mathrm{pk}}$ be a held-out pool of crossing targets (their designability under Ω_{pk} is generally unknown; the exact screen of §E removes only* provably*-undesignable targets, so an unsolved target may be a search failure* or *oracle-relative undesignability). A policy pair $(\pi_{\mathrm{before}}, \pi_{\mathrm{after}})$ exhibits an empirical crossing-family capability gain, adjudicated by the* single, fixed *crossing-aware oracle Ω_{pk}, if*

$$\mathcal{R}_{\Omega_{\mathrm{pk}}}(\pi_{\mathrm{before}}) \cap \mathcal{D}_{\mathrm{ho}} = \varnothing \qquad \textit{and} \qquad \mathcal{R}_{\Omega_{\mathrm{pk}}}(\pi_{\mathrm{after}}) \cap \mathcal{D}_{\mathrm{ho}} \neq \varnothing. \tag{8}$$

Both sides are scored by the same Ω_{pk}*, so the ascent is not an artifact of exchanging oracles; the decidable Prop. 2 gives the stronger, oracle-independent $\mathcal{R}_{\Omega_{\mathrm{cf}}}(\pi_{\mathrm{before}}) \cap \mathcal{L}_{\mathrm{pk}} = \varnothing$ as a pre-run bound. The empirical instance measured in §E is the left condition at $0/60$ (floor) against the right at* 3–$8/60$ *(panel operator).*

Definition 4 (Transferable crossing-family capability)**.** *The agent's* crossing capability *is a set $\Delta\mathcal{O} = \mathcal{O}_{\mathrm{after}} \setminus \mathcal{O}_{\mathrm{before}}$ of reusable components such that, with $\Delta\mathcal{O}$ enabled and the external pseudoknot solver disabled, $\mathcal{R}_{\Omega_{\mathrm{pk}}}(\pi_{\mathrm{after}})$ contains* held-out *targets of $\mathcal{L}_{\mathrm{pk}}$. It is* transferable *if this holds on targets never used to form $\Delta\mathcal{O}$. It is* autonomous relative to a tool class $\mathcal{S}$ *if it holds with every member of $\mathcal{S}$ disabled. Here $\mathcal{S}$ = packaged inverse-design solvers; the crossing-aware folding oracles are* not *in $\mathcal{S}$ and remain available; see Rem. 1).*

Remark 1 ("Solver-free" would have been the better word)**.** *In this paper "autonomous" means exactly one thing, that no packaged inverse-design solver is on the method path, and that is far weaker than the word carries in the agent literature. Our system queries crossing-aware folding oracles continuously and could not function without them. Where the distinction matters we write* solver-free*, and a reader who substitutes it everywhere loses nothing we intend. Two consequences follow. Before/after differ in feedback instrument as well as policy: the prior repertoire is scored through a pseudoknot-free channel and the after-policy queries crossing-aware predictors, so acquiring a new feedback instrument is part of what happened. And the entity credited with a gain is the oracle-augmented system under a declared protocol, not a policy in isolation, because the before/after step changes the policy, the in-loop objective, oracle access and the candidate-selection rule at once. Our design does not decompose that bundle, and §E shows a matched ablation failing to isolate any single component under the primary metric. Every "capability", "gain" and "new relative to the prior self" below is therefore a system-level empirical gain under augmented oracle access, and specifically not an identification of autonomous learning inside a policy. The single-oracle-versus-panel comparison runs at identical oracle access precisely so the crossing-aware channel is held fixed.*

Remark 2 (Component attribution strengthens the audit; it is not a necessary condition)**.** *We do not require that ablating $\Delta\mathcal{O}$ destroy the held-out rate. A component can be genuinely acquired and load-bearing as a bundle while no single element is individually necessary, so an all-or-nothing attribution test would reject real acquisitions. What we do instead is report which attributions we ran and how they came out, which for this study is mostly negatively: the agreement objective is decisive under r_{in} and not distinguishable from its deletion under r_{cons}; the decompose/compose scaffolding is inert under both; and the undirected control runs at a matched nominal cap rather than matched realized compute (§E, Table 12). Component attribution is therefore an optional strengthening, graded like the rest of the verdict. The audit's necessary conditions remain Eq.* (5)*, in which no per-component clause appears.*

Remark 3 (What we do and do not specify)**.** *We define the* audit *that recognizes a transferable, autonomous crossing capability; we do* not *here specify the algorithm that* produces *$\Delta\mathcal{O}$ (how the agent invents reusable components). That algorithm is the* object under test*: Def. 4 is exactly what distinguishes a genuine invented capability from a memorized instance (fails transfer) or a borrowed one (fails autonomy). This is the honest boundary between an instrument (ours) and a system (to be built and then measured). We cross that boundary in §E, where an LLM agent, given only primitives,* writes *a $\Delta\mathcal{O}$ that the audit then evaluates.*

D.1 Two Impossibilities, Kept Separate

The audit's negative side and undesignability are different statements and must not be conflated. Confinement (Prop. 2) is about the repertoire: $\mathcal{R}_{\Omega_{\mathrm{cf}}}(\pi_{\mathrm{before}})$ excludes T because the CF oracle

cannot represent it. Undesignability (Zhou et al., 2024) is about the target and energy model: no sequence realizes T under a criterion. It is provable by rival-structure construction: from the target's motifs one exhibits a rival at least as favorable as T for every compatible sequence, so T is never the unique optimum (Zhou et al., 2024; 2025). Since deciding "is T designable?" is NP-hard in general (Bonnet et al., 2020), a positive undesignability certificate, when the construction succeeds, is a valuable decidable witness on an otherwise hard question. Undesignability binds the ascended agent too: no repertoire realizes a target that no sequence realizes. The ascent's negative side uses confinement, not undesignability.

An optional, distinct capability. Undesignability enables a separate two-sided predictive capability: per held-out target the agent commits either a design or an undesignable claim, adjudicated respectively by the oracle panel or by an undesignability certificate. We delineate it from the crossing axis and do not pursue it here, noting only that a committed undesignability claim is a computation under sealed commitment, not a hypothesis in the usual empirical sense.

E Experiments: Measuring Oracle-Gaming, and Auditing the Claim It Inflates

This section does three things. It *measures* how far a single fallible oracle can inflate a capability claim: an invented, solver-free operator reaches $43/60$ under the predictor it optimizes and $1/60$ under three, with the discriminating comparison paired against an external solver that optimizes something else. It *audits* the system we then built to survive that measurement, reporting each component of the evidence vector separately, including the two it does not establish. And it establishes that the instrument *discriminates*, by measuring an arm that clears the bar our own does not. That is where this section's one clearly positive result lives, and why it is not filed as a supplement. The three arguments are kept in separate subsections, because they are answerable separately and a reader may want only one. §E.1 is the measurement, §E.2 the audit of our own operator, §E.3 the positive control and the seven interventional attribution tests, §E.4 what the target sets can support, and §E.5 the statistical conventions and remaining robustness checks. Everything is on the n=60 Pseudobase++ crossing pool unless a larger or structurally disjoint pool is named; every rate is one of the three metrics of Eq. (6), never a blended "panel" number.

Setup: an exact difficulty axis, a validated panel, and an exact designability screen. *(i) The grade is intrinsic and exact.* Each target's family type and crossing-stem count are properties of the target, decidable in $O(|P|^2)$ and never inferred from whether the agent solves it. The graded frontier therefore carries no survivorship bias, and the boundary is the established CF $\rightarrow$ MCF line (§B.1), not a learned proxy. *(ii) Verifier validity is checked before verdicts are trusted.* The CF ladder and the pseudoknot panel are reused instruments (Anonymous, 2026); we run the specificity half here (skeleton designs, expected zero crossing recall) and inherit native recall from that companion. Without it a lenient panel could credit skeletons as pseudoknots and manufacture an ascent. *(iii) Undesignable targets are excluded only on an exact certificate*, by rival-structure construction (Zhou et al., 2024; 2025) and never on solver failure, so hard-but-designable targets stay in; on the primary pool it excludes none. We also do not infer designability from native provenance, since our calibration shows $\Omega(x_{\mathrm{nat}}) = T$ usually fails ($\sim 1\%$ native recall). Oracle-relative designability of the pool is therefore *unknown*, and an unsolved target may be a search failure or an oracle-relative impossibility.

Positive-side adjudication under oracle-dependence. Because the pseudoknot verdict is oracle-dependent (§B), a target counts as realized only under an explicit rule: (i) mixed-panel-consistent, the primary rate; (ii) reference (pKiss); and (iii) envelope $[\min, \max]$ across predictors, reported alongside. We fix two acceptance predicates explicitly, because conflating them is the classic pseudoknot-evaluation error. Writing $P(\cdot)$ for a pair set and $P_{\mathrm{nest}}(\cdot)$ for its *nested projection* (crossing pairs deleted, which is what a pseudoknot-free parser sees),

$$A_{\mathrm{full}}(x, T) = \mathbf{1}[P(\Omega(x)) = P(T)]\,, \qquad A_{\mathrm{nest}}(x, T) = \mathbf{1}[P_{\mathrm{nest}}(\Omega(x)) = P_{\mathrm{nest}}(T)]\,. \tag{9}$$

Every rate in this paper uses A_{full}. A predictor confirms iff its predicted pair set equals the target's *entire* pair set — nested *and* crossing pairs, after bracket-type normalization, with no extra or

missing pairs (`forms_target`). The mixed panel requires this of all three. A_{nest} is reported only as the *blind rule whose false accepts we count* (Table 10): for a crossing target $T \in \mathcal{L}_{\text{pk}}$, Prop. 2 makes $A_{\text{full}}(x, T) = 0$ under Ω_{cf} for every x, whereas A_{nest} can and does return 1. So A_{nest} acceptances are evaluator errors, never evidence that a target needs no crossing. (The $\sim 1\%$ native recall of §C.1 is measured under a *laxer* crossing-pair-*recovery* predicate, so the panel's recall under A_{full} is no higher.) Disagreements trigger abstention; we never treat a single predictor as ground truth.

E.1 The Measurement: One Fallible Oracle Credits 43 Designs, Three Credit 1

The naive operator crosses only by gaming the oracle. We instantiate the full audit on the $n = 60$ development/evaluation pool with an invented operator, PKDECOMP: decompose the target into its context-free skeleton, sub-solve it with a pseudoknot-free designer, compose the crossing stem by complementarity, and repair against the pseudoknot oracle — the one move the {REMC, DesiRNA} toolbox never makes. In its first version PKDECOMP repairs against *pKiss alone*; we call that version the **single-oracle operator**, and refer to it by that name throughout. It is the arm the audit convicts of oracle-gaming. It never calls an external pseudoknot solver, so its autonomy is structural. Under the single reference oracle (pKiss) the outcome looks like a decisive ascent: $43/60$ solved, matching the external SOTA solver while calling nothing, far above the $0/60$ floor. The negative side is firm on both its components: the Ω_{cf}-certification bound is exact (Prop. 2) and the empirical CF-confined floor is $0/60$, with every skeleton design a naive-MFE-style false positive the panel refutes (a $0/60$ panel credit on the E0 skeleton control below).

The positive side does not survive the panel. The operator optimized pKiss in the loop; re-adjudicated by a held-out predictor, only $2/43$ of its solves are confirmed and one is panel-unanimous (Table 10). The apparent $43/60$ ascent collapses to $1/60$ (Fig. 8): an oracle artifact, the operator having learned the blind spots of the predictor it was scored on (Skalse et al., 2022; Gao et al., 2023). A single-oracle evaluation would have reported a new autonomous capability; under the abstaining panel (§C.1) almost nothing survives. The gap between 43 and 1 is what a one-sided criterion cannot see.

What the ratio is, and what carries the claim. We name the collapse by its ratio because that is the quantity a practitioner cares about. But the ratio's denominator is a single target, and we will not pretend otherwise. $1/60$ has an exact binomial interval of $[0.0004, 0.089]$ against $43/60$'s $[0.586, 0.825]$, so the data are compatible with an inflation anywhere from roughly $7\times$ upward. Had two designs survived instead of one it would read 21.5-fold. **Nothing in this paper rests on the ratio.** What carries the finding is the paired contrast below, on the same 43 targets: 2 confirmed against 26, exact McNemar $p{=}8.1 \times 10^{-7}$. Its uncertainty is small because it is paired and its denominator is 43, not 1.

Oracle-gaming vs. oracle-robustness: a contrast with the borrowed solver. Is the collapse specific to our operator? Re-adjudicating the external solver's 43 pKiss-solves under the same panel (Table 11): DesiRNA optimizes minimum free energy, not the pseudoknot oracle, and $26/43$ of its solves survive ProbKnot, whereas our operator, which optimized pKiss, retains only $2/43$.

Is that a paired comparison? Two "$43/60$" figures need not name the same 43 targets, in which case the ratio would mix method with target composition and difficulty. We checked (`contrast_paired.py`), and the check comes out cleanly. The two pKiss-solved sets are *identical*: intersection 43, Jaccard 1.000, neither arm solving a target the other misses. The comparison is therefore exactly paired by construction, on targets with an identical difficulty profile (mean crossing-pair count 5.26 for both, against a pool mean of 5.48). On that common subset the paired McNemar is $b{=}25$ (DesiRNA only), $c{=}1$ (ours only), exact $p{=}8.1 \times 10^{-7}$, paired risk difference $+0.558$ with bootstrap 95% CI $[+0.395, +0.721]$; at the composition-free full $n{=}60$ denominator it is $26/60$ vs. $2/60$, $+0.400$ $[+0.267, +0.533]$. So target composition is excluded as an explanation.

What that still does not license. The two arms are two different methods, not a randomized manipulation of one method's training oracle, so the design is observational in the arm. pKiss-training is confounded with every other difference between DesiRNA and our operator: search algorithm, energy model, candidate quality, budget. The defensible reading is therefore *association*, not causal localization: the result is *consistent with substantially stronger oracle-specific over-optimization by*

Table 10: **Cross-oracle panel adjudication of the invented operator's 43 pKiss-solves.** Read the top block down the count column: $43 \to 2 \to 1$ as adjudication moves from the oracle the operator optimized to predictors it never saw. The lower block is the exclusion side, and its two numbers are *not* confirmations of the design: all 43 targets are crossing (3–9 crossing pairs each), so a pseudoknot-free fold realizes none, and the 4 counts how often the pseudoknot-blind predicate would have been fooled.

adjudicator / predicate	count / 43	what the number counts
crossing-aware adjudicators: the collapse		
pKiss, A_{full}	$43/43$	the in-loop oracle, the rate the operator was optimized to produce
ProbKnot, A_{full}	$2/43$	first crossing-aware predictor the single-oracle operator never saw (in-loop for the panel operator)
all three predictors, A_{full}	$\mathbf{1/43}$	mixed panel, 2 in-loop + 1 held out; $1/60$ of the pool, the audited outcome
pseudoknot-free adjudicators: the exclusion side, not confirmations of the design		
pseudoknot-free MFE, A_{full}	$\mathbf{0/43}$	a theorem, not a measurement (Prop. 2: $\operatorname{im}\Omega_{\text{cf}} \subseteq \mathcal{L}_{\text{nest}}$)
pseudoknot-free MFE, *blind* A_{nest}	$4/43$	designs the nested-projection rule would *falsely accept*: that rule's own error rate, not a property of the targets

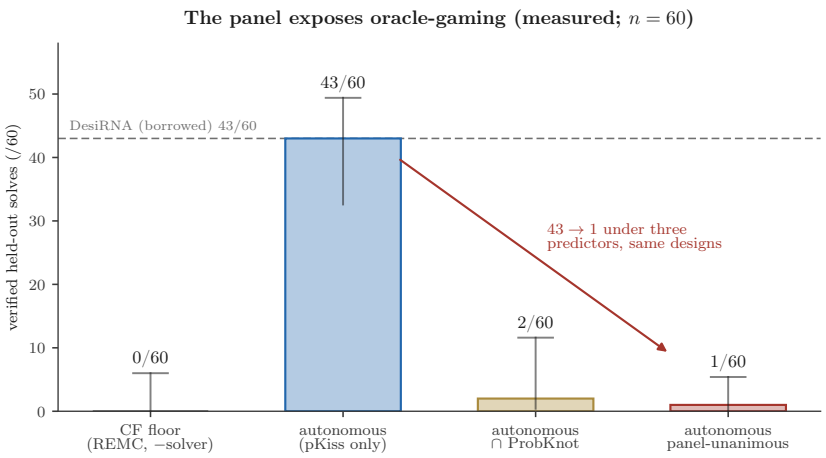

Figure 8: **The panel exposes oracle-gaming (measured, $n = 60$).** The invented operator's single-oracle solve rate ($43/60$, matching the borrowed DesiRNA ceiling) collapses to 2 under a held-out predictor and to 1 panel-unanimous, above the decidable context-free floor of $0/60$. Bars carry exact binomial intervals, and the collapse is annotated as $43 \to 1$ on the *same* designs, not as a "$43\times$" ratio: the last bar is a single design, so that quotient is unstable and nothing in this paper rests on it (§E). What a one-sided criterion cannot see is the gap, not its quotient.

the pKiss-trained method, and we do not claim it localizes gaming *to* that training. Two further caveats are worth stating. Under the full three-predictor unanimous panel neither arm is robust ($0/43$ for DesiRNA, $1/43$ for ours): ShapeKnots, run de novo without probing data, rejects almost everything, so the cleaner comparison is the two de-novo predictors (pKiss $\wedge$ ProbKnot), where the 2-vs-26 gap stands. And that even a SOTA solver is only $\sim 60\%$ cross-predictor-robust confirms the thesis: "forms the pseudoknot" is deeply model-relative.

E.2 Turning the Audit on Our Own Operator

It is a search, and the ablation says which part of it works. We do not pose this as capability *versus* search: the operator is a search, and the useful question is which ingredient carries the gain.

Table 11: **Oracle-gaming vs. cross-predictor carry-over** (each arm's 43 pKiss-solves, re-adjudicated). The pKiss-optimizing operator is $13\times$ less ProbKnot-robust than the MFE-optimizing external solver. The comparison is paired: the two pKiss-solved sets are the same 43 targets (Jaccard 1.000, matched mean crossing-pair count 5.26), so target composition cannot explain the gap. Paired McNemar b=25, c=1, exact p=8.1×10^{-7}, risk difference $+0.558$ $[+0.395, +0.721]$ (`contrast_paired.py`). Because the arms are two different methods rather than a randomized manipulation of one method's training oracle, we read this as an *association*. Under the full unanimous panel neither arm is robust.

method (objective)	**pKiss**	**ProbKnot**	**ShapeKnots**	**unanimous**	**ProbKnot / 60**
DesiRNA (minimum free energy)	43/43	**26/43**	1/43	0/43	26/60
invented operator (pKiss in-loop)	43/43	**2/43**	3/43	1/43	2/60
paired, same 43 *targets*: b=25, c=1, exact McNemar p=8.1×10^{-7}, $\widehat{\Delta}$=$+0.558$ $[+0.395, +0.721]$					

We answer it in the bit-reproducible regime only (Table 12; App. G gives the fix, and the earlier wall-clock ablations it supersedes are not reported, since every conclusion below is stronger without them). The design is five arms of the same operator at an identical 150-call pKiss allowance, each differing from the full operator in exactly one component, run at five noise-free seeds and paired on target and seed.

Three findings come out of it. All three are sharp, and only the first is flattering to us. *(i) The repair loop is the operator.* Delete it and r_{in} falls $37.0 \rightarrow 1.0$ (Δ=$+0.600$ $[+0.487, +0.707]$) and r_{cons} to 0. The crossing capability is stochastic search against the pseudoknot oracle, which is precisely why it games: search against a scorer finds its blind spots. *(ii) The objective is load-bearing only on the metric it optimizes.* Deleting it costs $+0.140$ $[+0.063, +0.220]$ on r_{in} and *exactly zero* on r_{cons}: $+0.0000$, $[-0.060, +0.050]$, on 300 paired cells. With execution noise removed this is a measured absence, not an underpowered null. So "optimize agreement" is supported as a *choice of success predicate* and not as a search objective that earns its keep. *(iii) The* CF *decomposition is load-bearing after all*, $+0.093$ $[+0.023, +0.173]$ on r_{in}, where the noisier estimate had it inert: the one place removing execution noise reversed a conclusion, and we flag it instead of quietly adopting the new sign.

One matching caveat survives determinism and cannot be removed. Equal *allowance* is not equal *use*: the full operator often solves and stops, so realized pKiss calls run 52.6–105.5 across arms (Table 12, last column). Equalizing realized use would mean penalizing success. The ablated arms consume *more* compute, so their lower r_{in} is not a budget artifact; the comparison is nonetheless matched on access, not on use.

How much does (ii) actually say? Less than the interval suggests, and we would rather state that than let a reader find it. r_{in} is pKiss $\wedge$ ProbKnot and the undirected walk optimizes neither, so "deleting the objective costs $+0.140$ on r_{in}" is close to tautological: the arm fails the conjunct it stopped pursuing. The informative half of (ii) is the other number, the *exact* zero on r_{cons}, which is not tautological, because r_{cons} adds a predictor neither arm optimizes, and because a two-sided interval of $[-0.060, +0.050]$ on 300 paired noise-free cells is a measurement rather than a failure to find one.

What this licenses about the success predicate, stated precisely. The natural conclusion, that the success predicate and not the search is the causal lever, is not supported in that form. Changing the predicate from one predictor to a two-predictor conjunction is *associated* with a higher r_{cons} in the single-oracle-versus-panel comparison, while deleting the objective from the operator does not lower r_{cons} at all. Those reconcile only if something other than the objective distinguishes the two systems, and this design cannot say what. We report the association and stop.

The uncomfortable reading, stated in our own words. Put (i) and (ii) together and the description of our operator that the data actually supports is: **a stochastic search against a fallible oracle, with a better success predicate and no better guidance**. It is not a design principle, and we prefer to write that sentence ourselves than to shelter behind "a search procedure can itself be a capability", which is true but is not what these numbers show. It also sharpens what the positive control is for. The agent-written operators clear the *same* predicate on the *same* targets, so the predicate is held

Table 12: **Component ablation and the headline comparison, bit-reproducibly at equal allowance**: 5 nominal seeds, one noise-free run each, single-oracle operator included. Counted budgets replace wall-clock caps and VIENNARNA's generator is seeded per attempt, so a seed fixes the run bit-for-bit; every arm gets the same 150-call pKiss allowance. Δ is the paired difference against the reference with a two-level bootstrap over targets and seeds (B=20,000); **bold** excludes zero. r_{in} is each arm's *own* predicate, pKiss alone for the single-oracle operator, so it is not comparable across those groups; r_{cons} is the cross-arm metric. "real pKiss" is realized calls per target, which differ because the full operator often solves and stops. (`panel_deterministic.sh`, `det_stats.py`.)

	r_{in} (own predicate)		r_{cons} (all three)		
arm	**mean/60**	Δ [CI]	**mean/60**	Δ [CI]	**real pKiss**
single-oracle operator	42.6	*n/c*	0.00	**ref**	52.6
full operator (reference)	37.0	—	4.00	—	82.2
headline: full − single-oracle on $r_{\text{cons}} = +.\mathbf{067}$ $[+.023, +.120]$; single-oracle $0/60$ at all five seeds					
− objective (undirected)	28.6	$+.\mathbf{140}$ $[+.063, +.220]$	4.00	$+.000$ $[-.060, +.050]$	99.5
− objective, − bias (blind)	25.4	$+.\mathbf{193}$ $[+.107, +.287]$	2.40	$+.027$ $[-.017, +.077]$	105.5
− repair loop entirely	1.0	$+.\mathbf{600}$ $[+.487, +.707]$	0.00	$+.\mathbf{067}$ $[+.023, +.123]$	1.0
− CF decomposition	31.4	$+.\mathbf{093}$ $[+.023, +.173]$	3.40	$+.010$ $[-.033, +.050]$	100.1
− complementarity composition	34.4	$+.043$ $[-.030, +.137]$	5.40	$-.023$ $[-.080, +.030]$	90.3

fixed between them and us; whatever separates them is in the search, which is exactly the dimension our own ablation says we did not move.

The panel-optimizing operator, and its held-out rate. The operator whose repair loop optimizes agreement between pKiss and ProbKnot solves 34–37/60 on its own two oracles across five seeds, against the single-oracle operator's ProbKnot-robust 2/60. Of those, 3–8 (mean 5.0) are also panel-unanimous, against $\sim 1/60$ for the single-oracle operator and $0/60$ for the SOTA solver. Read every band of this form as the range of the runs named beside it: pooling all 24 executions gives 30–39/60 and 1–9/60 (mean 3.29). Panel-unanimity puts two of its three conjuncts inside the operator's objective, so it cannot by itself show generalization. On the one genuinely held-out predictor the operator is higher at every seed but significant in one of three (n=125 ID-disjoint targets; $\widehat{\Delta} = +0.008, +0.056, +0.064$). We record that as *weak* and do not build on it. In *absolute* terms it is not small once given its reference class: the same adjudicator recovers only 11.3% of native pseudoknots from their own sequences and confirms our designs at 8.0%, about 70% of its own ceiling, against the single-oracle operator's 3.7% (App. H.2).

Δ_{single}: separated, but by how much depends on the estimator, and we say so. Run on the *same* seeds and targets as the operator, the single-oracle operator is panel-unanimous on $\{1, 0, 0, 0, 0\}/60$ against $\{8, 3, 5, 5, 4\}/60$. Per seed the paired McNemar has c=0: the single-oracle operator never solves a target the operator misses. **The quantity is estimated four times in this paper and the point estimate falls every time the design is tightened**: $+0.080$, $+0.056$, $+0.048$ as execution replication is added, then $+0.067$ $[+0.023, +0.120]$ in the bit-reproducible regime with the single-oracle operator at $0/60$ at every seed (Table 12). App. G gives each design, its intervals, and the conservative one-execution-per-seed variant that reaches 0.000. Three readings, and only the first is comfortable. Determinism removes the *execution* level of variation but not the *target-set* level the decline pointed at, and its narrower interval reflects one fewer variance component, not more evidence. The most robust statement uses no interval at all (Fig. 9): the operator is strictly higher in 14 of 15 independent executions and 15 of 15 on the structurally disjoint pool. And $+0.067$ sits at 77% power for this design (§E), i.e. at its resolution limit, which is what "established, narrowly" means in units a reader can check. What is settled is the sign; the magnitude is not, and the search-free probe below removes most of it on a different ground.

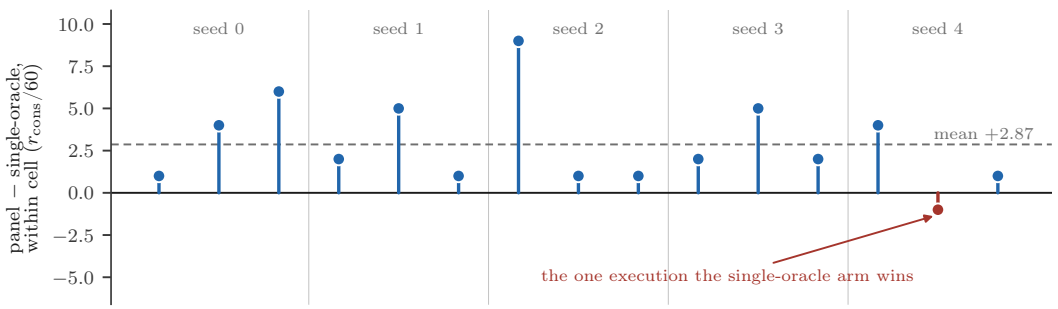

Figure 9: **The statement about the headline effect that uses no interval at all.** Within-cell paired difference in r_{cons}, panel operator minus single-oracle operator, for each of the 15 wall-clock executions (5 nominal seeds $\times$ 3 executions) on the 60 common targets. Fourteen point one way and one points the other. This is the form the headline result is most defensible in, because it assumes nothing about the resampling unit and nothing about which variance components a bootstrap should nest: it is a sign test on independent executions, and that is why §E.2 settles the sign while declining to settle the magnitude. Bar heights are the per-cell McNemar discordance $b - c$, parsed from `gvp_stats.txt`.

Two confounds this comparison exposes rather than removes. The arms are matched on oracle *access*, not on oracle *use*. Per target the operator makes 128.0 pKiss and 63.5 ProbKnot calls, against the single-oracle operator's 55.2 and 0, because that arm's single- predictor objective is satisfied at $43/60$ and stops. Realized compute is a *mediator* of the objective change, so Δ_{single} measures "better designs" and "searches $2.3\times$ longer" summed, and this design cannot split them. And it is a between-*system* contrast: two different methods, not a randomized manipulation of one method's oracle.

r_{cons} is not metric-neutral, so we redo the comparison on a statistic that is. $r_{\text{cons}} = r_{\text{in}} \wedge$ ShapeKnots, and r_{in} is *the panel operator's own success predicate*. Scoring both arms on r_{cons} therefore scores one of them on its objective plus one term, which by our own argument in §C.1 cannot by itself evidence generalization. The information r_{cons} adds over r_{in} is the conditional term, so we make that term the statistic:

$$C(\text{arm}) \;=\; \Pr\big[\,\text{a predictor never in } \textit{that arm's} \text{ loop confirms the design} \;\big|\; \text{the arm's own success predicate is satisfied}\,\big]. \tag{10}$$

C is neutral because every arm is scored on how far *its own* successes carry to a predictor *it* never optimized, with its own predicate as the denominator. ShapeKnots is in no arm's loop, so carry-over to ShapeKnots is defined identically for all of them (Fig. 10). Three readings follow, and they do not all point the same way.

The single-oracle operator comparison survives the change of metric. Difficulty-matched to the 528 (target, seed, execution) cells where both arms satisfy their own predicate, ShapeKnots confirms 48 of the panel operator's designs against 22 of the single-oracle operator's (b=41, c=15, exact p=6.9×10^{-4}, difference $+0.049$). That the neutral statistic reproduces the r_{cons} value of $+0.048$ almost exactly is reassuring rather than circular: the two are computed from different denominators. One conditioning does *not* agree and we report it: if both arms are instead held to the panel operator's bar (r_{in} for both), the single-oracle operator's rate is nominally higher ($5/32$ against $48/530$). That comparison conditions the single-oracle operator on having passed ProbKnot — the very carry-over at issue — and retains 32 of its 643 designs, so we do not treat it as primary. But with an interval of $[0.053, 0.328]$ it does not refute the primary reading either, and a reader who prefers it should read Δ_{single} as unestablished.

The objective ablations do not survive it. The undirected walk and the oracle-blind walk share the panel operator's success predicate and differ only in guidance, and on C they are indistinguishable from it (0.095 and 0.110 against 0.094; paired p=0.57 and p=0.29, both point estimates *against* the full operator). So what the two-oracle conjunction gains comes from the *predicate*, not from the search that pursues it. That is the same conclusion $\Delta_{\text{undirected}}$ reached on r_{cons}, now on a metric that cannot be accused of favouring either arm.

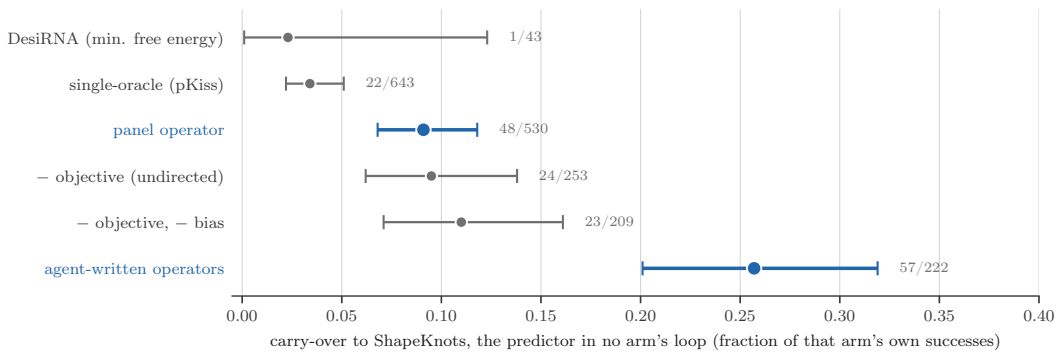

Figure 10: **Conditional carry-over C (Eq. (10)): the metric-neutral comparison.** Each arm is scored on the fraction of its own successes that a predictor never in its loop confirms, so the denominators differ by design and that is what makes the statistic neutral. Bars are exact binomial intervals; counts are successes / own successes. The two coloured rows are the comparison the figure exists to make: the hand-built panel operator against the frozen agent-written operators of App. J, which optimize the same predicate. The ProbKnot direction is meaningful only for the two arms that do not optimize it (Table 11). The agent row pools six single runs; its two zero-failure models replicated at 5×3 give 0.302 $[0.275, 0.330]$.

Cross-predictor robustness is pairwise, not global. DesiRNA carries over to ProbKnot at $26/43$ against our single-oracle operator's $2/43$ (§E, Table 11), and to ShapeKnots at $1/43$, the *lowest* rate of any arm in Fig. 10. An arm can be $13\times$ more robust under one un-optimized predictor and the least robust under another. ProbKnot and DesiRNA's minimum-free-energy objective are both built on the same nearest-neighbour thermodynamic parameters, and ShapeKnots run de novo is not; the natural reading is that "robustness" here tracks shared model family, not correctness. That is a limitation of the whole panel, not of one arm, and it is the strongest argument in this paper for adding a predictor from a different model class (§6).

E.3 Does the Instrument Discriminate? A Positive Control That Becomes a Result

Does the instrument discriminate, or is it merely insensitive? A positive control. We have said more than once that nothing we build saturates the panel and offered that as evidence of stringency. It is not: an audit nothing passes and an audit that cannot detect anything produce the same table, and at $\sim 1\%$ recall on natives insensitivity is the live alternative. Separating them needs an arm that *does* clear the bar, measured identically. We already had one, filed as a supplement. The frozen LLM-written operators of App. J optimize pKiss $\wedge$ ProbKnot, the panel operator's own predicate, so ShapeKnots is held out from them in the same sense. Figure 11 is the loop that produced them, which ran to completion before this control was conceived of and was never tuned against the held-out predictor.

Replicated at the depth of the headline experiment (the two operators with no runtime failures, frozen code re-executed at 5 seeds $\times$ 3 executions, `agent_replicate.py`), their carry-over is $333/1104 = \mathbf{0.302}$ $[0.275, 0.330]$ against the hand-built operator's $48/530 = 0.091$ $[0.068, 0.118]$; the intervals do not overlap. The three executions inside a seed are *bit-identical* in $5/5$ seeds, so unlike our own operator these are reproducible runs.

Paired, at the standard we impose on the transplant arms. Those are two conditional rates on two different denominators, and Fig. 12's caption forbids exactly that reading for every other arm. We apply the same standard here, not a weaker one, to the arm that carries the paper's discriminating-power claim. Both arms optimize the same pKiss $\wedge$ ProbKnot predicate and are scored by ShapeKnots, held out of both, so restricting to the (target, cell) units where *both* arms cleared that shared bar gives a paired comparison. Same target, same nominal seed, same execution index, same in-loop bar met; the only open question is whether the held-out predictor confirms. On those n=951 units the agent operators carry over at 0.293 against 0.095, with b=218 discordant pairs favouring the agents against c=29 favouring us: a paired difference of $+0.199$. Table 13 splits that pooled

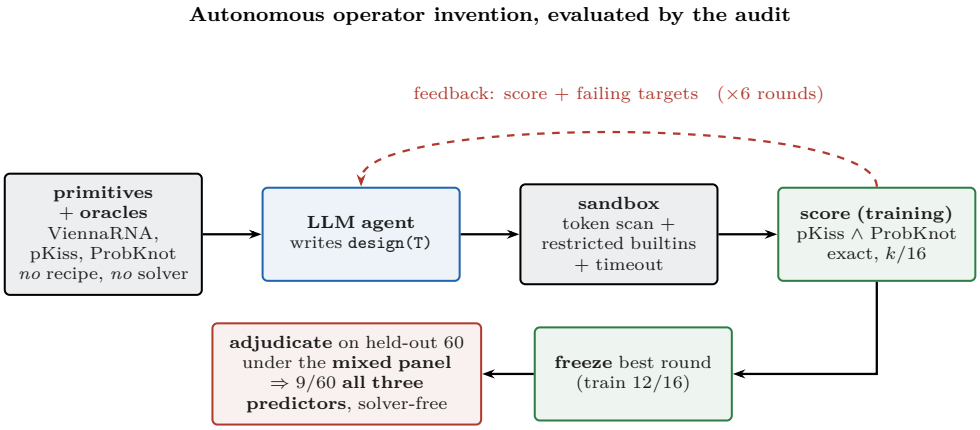

Figure 11: **How the agent-written operators were produced** (App. J gives the protocol in full). A frontier model is given primitives — ViennaRNA, pKiss, ProbKnot — with no recipe and no packaged inverse-design solver, and writes a `design(T)` function. The code runs in a sandbox under a token scan, a restricted-builtin whitelist and a timeout; it is scored on 16 *training* targets by pKiss ∧ ProbKnot exact match; the score and the failing targets are returned as the only feedback, for six rounds. The best round (12/16) is then **frozen** and adjudicated once on the held-out 60 under the full panel, where it reaches 9/60 under all three predictors. Two things matter for the control this section runs. The training signal never contains ShapeKnots, so the held-out predictor is held out of the *invention* loop and not merely the repair loop. And the freeze happens before any held-out number exists, so the round was not chosen on the quantity we then report.

Table 13: **The positive control, model by model** (`arep_matched.py`, `agent_budget.py`; 5 seeds × 3 executions per arm). Each agent-written operator is compared with our hand-built operator on the (target, cell) units where *both* cleared the shared in-loop bar (pKiss ∧ ProbKnot), and scored by ShapeKnots, held out of both loops. The hand-built column therefore changes between rows: it is the same operator re-scored on each row's matched units, which is what makes the difference paired. Intervals and p-values resample and permute whole *targets*, the correlated unit; the exact McNemar on these counts is 3.6×10^{-26}, 6.4×10^{-13} and 5.1×10^{-37} respectively, and we print those only to show the size of the correction we are applying, never as the design's p. The last column is realized crossing-oracle load per target in the same 20 s regime, against the hand-built operator's 128.0: both agents win while searching several times *less*, which is the R2 rung obtained in the direction opposite to the worry.

operator	**matched** n	**agent** carry-over	**hand-built** same units	**paired Δ** [95% clustered]	**cluster** perm. p	**pKiss** /target
gemini-3.1-pro	442	$161/442 = 0.364$	$45/442 = 0.102$	$+0.262$ $[+0.151, +0.382]$	1×10^{-4}	27.8
kimi-k3	509	$118/509 = 0.232$	$45/509 = 0.088$	$+0.143$ $[+0.059, +0.241]$	2.2×10^{-3}	13.1
pooled	**951**	$\mathbf{279/951 = 0.293}$	$90/951 = 0.095$	$\mathbf{+0.199}$ $[+0.108, +0.297]$	$\mathbf{5\times10^{-5}}$	—

discordant pairs (agent only / hand-built only): 127/11, 91/18, pooled 218/29; 41–42 target clusters
hand-built reference: 128.0 pKiss calls/target in the same 20 s regime; the compute control reaches 196.2

figure by model, because a result carried by one lucky operator and a result reproduced by two independently written ones are different claims, and only the second is worth reporting.

We do not quote the exact McNemar these counts give. It is 5.1×10^{-37}, and it assumes 951 independent units when they come from 60 targets contributing ∼16 correlated observations each. Fig. 12's caption rules out that anticonservatism for every transplant arm, so applying it to the arm carrying this paper's positive result would be indefensible. Resampling whole *targets* gives $[+0.108, +0.297]$ over 42 clusters; a permutation test flipping each target's discordant pairs together gives $p{=}5\times10^{-5}$. Per model the clustered intervals are $[+0.151, +0.382]$ and $[+0.059, +0.241]$. The correction moves

the p-value by 32 orders of magnitude and changes no conclusion. Matching costs the point estimate almost nothing (0.302 to 0.293), which is informative: the advantage was not an artifact of the two arms solving different targets (`arep_matched.py`). One quantifier the matching does not repair: these are the 2 of 6 invented operators with no runtime failures, a model-level survivorship selection on top of the target-level one, so the supported claim is that *some* agent-written operators carry over better.

How much compute do the agents actually spend? We stopped arguing from their source and counted. The allowance control above is only informative if it reaches the regime it is meant to rule out, and the frozen sources make that doubtful: one asks for 15 restarts $\times$ 3000 iterations against our 6×800. But a constant in the source is what the code *requests*, not what it spends under the 20 s per-target cap the replication imposed, so we instrumented the two crossing-aware primitives and re-ran the frozen code in that regime (`agent_budget.py`). The measurement inverts the worry. Realized pKiss calls per target are ≈ 28 for gemini-3.1-pro and ≈ 13 for kimi-k3, against the hand-built operator's $\mathbf{128.0}$ in the same regime: $4.6\times$ and $9.8\times$ *fewer*, with the $3\times$-allowance control sitting $7\times$ above the hungriest of them. This licenses a stronger statement than "not bought with compute": **the agent operators reach** 0.293 **carry-over while spending a fraction of the oracle calls of the operator they beat at** 0.095. (Two independent runs of the instrumented measurement gave $28.6/12.9$ and $27.8/13.1$; our harness runs it in-process and serially, so it does not reproduce the fork-per-call isolation of the replication, and the small spread is that. It does not touch the conclusion.)

The inverse control, and the compute-matched comparison it buys. The measurement also generated a hypothesis and let us kill it in the same pass. If the agents are not searching harder, perhaps they benefit from searching *less*. kimi-k3's source contains an explicit `budget = [30]`, a self-imposed oracle-call cap, and a search forced to commit early might accept candidates from a different region of sequence space. We ran the compute control the other way: the hand-built operator at a 30-call pKiss allowance, nothing else changed, deterministic, five seeds (`tinypk.sh`, `tinyctl_stats.py`). Carry-over is $8/75 = \mathbf{0.107}$ $[0.030, 0.203]$ against the baseline's 0.108, while r_{in} collapses from 37.0 to 15.0 per 60. By the rule fixed before the numbers, that a mechanism counts only if carry-over rises *and* r_{in} holds, that is a fifth null, and a clean one: the rate did not even inflate in the artifactual way a halved denominator invites.

That arm is worth more than the hypothesis it refutes, because it realizes 26.0 pKiss calls per target against gemini-3.1-pro's 27.8. **At matched realized compute, within** 7% **on the coordinate that matters, the hand-built operator carries over at** 0.107 **and the agent-written one at** 0.367**.** This discharges the compute-matched (rather than cap-matched) control the previous revision listed as owed, and it closes the compute question in both directions: the advantage does not appear when we are given three times the allowance, and it does not appear when we are held to the agents' own.

A structural property of the designs, gated then transplanted. Composition and compute are excluded, so the surviving hypothesis is that the accepted designs differ in *where* their nucleotides sit relative to the target's pair set. We pre-stated six such features and two gates: (a) the feature differs between the arms, and (b) it predicts carry-over *within* the hand-built arm, where the program is held fixed. Overall GC went in as a negative control, and earned its place at once by passing gate (b) outright (AUC 0.749, $p{=}1.9 \times 10^{-10}$), so an unadjusted gate admits anything GC-correlated. With both gates adjusted for GC by logistic regression and a permuted feature as the adjusted null ($p{=}0.813$), `gc_cross` ($p{=}0.558$) and `a_loop` ($p{=}0.176$) fall to fingerprints, and two features are undefined on 513 of 530 designs because a bare H-type pseudoknot has no non-crossing pairs. One survives: `decoy`, the count of off-target windows complementary to the crossing stem's $5'$ arm, of which agent designs carry 0.003 each against our 0.060.

That earned a transplant, and the transplant refutes it. **Our first attempt does not count and we say so**: adding `decoy` as an acceptance tiebreak left the accepted designs' decoy count unchanged ($0.0757 \to 0.0761$), and by the standard of Fig. 12 a transplant that changes nothing measurable is a bug. Moving the intervention to the step that determines the feature, choosing the crossing stem's bases to minimize decoys and still calling no oracle, engages properly: decoys fall to 0.0452 and the designs carrying any halve from 12 to 6. Carry-over then goes the wrong way, $16/177 = 0.090$ $[0.034, 0.157]$ against the baseline's 0.108, with a paired difference of -0.031 on the 162 cells both arms solved ($p{=}0.33$) and r_{in} at $35.4/60$.

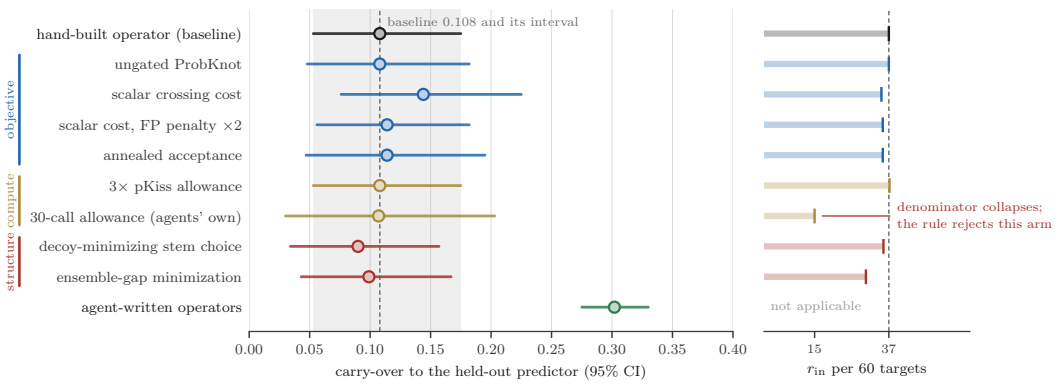

Figure 12: **Nine interventions on the hand-built operator, against the arm they are trying to reproduce.** Each row is that operator with exactly one thing changed, run deterministically at 5 seeds. Carry-over is $P(\text{ShapeKnots confirms} \mid \text{the arm's own pKiss} \wedge \text{ProbKnot predicate confirmed})$; intervals are percentile bootstraps resampling *targets*, the resampling unit throughout, because units inside a target are correlated. Three standards are fixed here and invoked by reference elsewhere. (i) Two conditional rates on different denominators are not a comparison, so every cross-arm claim is made paired on the cells both arms solved. (ii) An independence-assuming p-value on these units is anticonservative and is not quoted. (iii) A mechanism counts only if carry-over rises *and* r_{in} holds, since a rate bought by collapsing its own denominator is not an improvement; the 30-call arm is the case that rule exists to catch, and a transplant that changes nothing measurable is a bug rather than a null. Every interval contains the baseline's and none reaches the agents' lower bound of 0.275. Exact counts are in the committed outputs the panels are drawn from (`attr_stats.txt`, `tinyctl_stats.txt`, `ndc_stats.txt`, `dom_stats.txt`).

A seventh test, distributional rather than scalar. A reviewer asked the sharpest remaining question: a difference could live in the *whole distribution* of accepted designs while every hand-picked scalar came back null. Answering it took three designs and the first two failed their own controls. Unstratified, all six pre-stated quantities came back significant, the signature of a test measuring composition six times over. GC-decile stratification did not repair it: the negative control, GC against itself within GC bins, came back significant. Caliper-matching on GC (± 0.01, no replacement, $399/530$ pairs) finally calibrates it, with the control at $p{=}0.98$ and matched mean GC 0.4643 against 0.4640. **And a difference survives at matched composition**: the agent designs' ensemble gap $(\text{MFE} - \text{EFE})/n$ is 0.005 against our 0.012 (KS $D{=}0.466$), with longer unpaired loops (max 13.9 against 10.8). They concentrate the Boltzmann ensemble on a single fold, and a dominant fold is one any model confirms (`d3_distributional.py`).

That is a mechanism-shaped lead, and the transplant kills it. Minimizing the ensemble gap where it is determined, using a nested partition function outside every arm's objective and outside the counted allowance, engages properly ($0.0126 \rightarrow 0.0080$ per design, 60% of the way to the agents' value). Carry-over does not follow: $15/151 = 0.099$ $[0.043, 0.167]$ against 0.108, and on the 141 cells both arms solved the paired difference is **exactly zero** ($b{=}7$, $c{=}7$, $p{=}1.0$), with r_{in} falling $37.0 \rightarrow 30.2$ (`dominant.sh`, `dom_stats.py`). That leaves two independent leads, both composition-controlled and both dead under intervention.

What seven interventional tests bought, which is not what we set out to buy. The mechanism hunt failed: the advantage is robust, is not compute in either direction, is not legible in the source, and is not reproduced by either property that distinguishes the arms at matched composition. We had been reporting that as a confession. It is better read as the paper's most portable result, and reading it that way costs no evidence.

Four times in this study a statistically supported and mechanistically plausible attribution was wrong, and a control fixed in advance caught it. (i) `decoy` differs between the arms at $p{=}2.1{\times}10^{-4}$ and predicts carry-over *within* the hand-built arm at $p{=}2.6 \times 10^{-3}$ after adjusting for composition; biasing toward it moves carry-over from 0.108 to 0.107 and reverses the sign. (ii) The ensemble gap differs at matched composition with $D{=}0.466$ under a calibrated null and has a clean mechanism,

and biasing toward it gives a paired difference of *exactly* zero. (iii) Two successive designs of the distributional test were confounded by GC, and in both cases the negative control, not our judgement, said so. (iv) The agent source asks for 15 restarts $\times$ 3000 iterations against our 6×800, which reads as "they searched harder"; counting realized calls shows they spend 4.6–$10\times$ *fewer*.

The standard way to attribute a capability difference in a learned system is step one of what we did: find a property that differs between the systems and correlates with the outcome, then name it the mechanism. On this dataset that procedure would have produced two confident, well-powered, mutually independent false mechanisms, each with a publishable p-value, plus two further false conclusions from a plausible confound and a source constant. What separated them from real mechanisms was *intervening* on each and re-measuring under a rule fixed in advance. We therefore report the two-stage gate as a method rather than as our misfortune: observational screen, then transplant, with the transplant's *engagement* verified before its outcome is read. Its cost is one extra sweep per candidate; its benefit is that seven candidates are excluded by measurement, and no exclusion rests on our judgement about which properties matter.

None of this identifies the mechanism, and we do not soften that: a *discovery*-grade claim on this axis needs a transferable design principle, and one cannot transfer what one cannot identify. So we continue to claim no attribution, and no transfer either: on the structurally disjoint pool $9.7/27$ and $1.1/27$ targets error out per cell, leaving a denominator filtered towards easy targets, and rates there are recorded in App. J and used for nothing. Only two of six models had no runtime failures; the other four time out on 19–32 of 60 targets, and that exclusion is itself a selection we cannot fully audit.

Fairness requires putting these operators on the same axis we used to bound our own result, and they survive it better than we do. Of the panel operator's r_{cons} wins 85% fall on zero-margin targets; for the two replicated agent operators the figures are 76% and 63%, so on the 30 targets a random compatible sequence never reaches they take 45 and 54 wins against our 7. The advantage is therefore not an artifact of picking easier targets. It is *larger* on the stratum where search is actually required (`runs/cc_ascent/pool_margin.py`).

E.4 What the Target Sets Can and Cannot Support

How much of this is the method, and how much is which targets are easy? Every rate in this section is a property of a method *and* of a target set, and we had no way to separate them, because the only difficulty statistic we carried, the crossing-pair count, describes the target's topology and not how hard it is to hit. So we built a search-free probe. Draw 60 sequences merely *compatible* with T (Watson–Crick or GU at every paired position, uniform elsewhere), fold each under pKiss, and record

$$\Delta_{\mathrm{bp}}(T) \;=\; \min_{x \text{ compatible}} \big|\, \mathrm{pKiss}(x) \mathbin{\triangle} P(T) \,\big| \;/\; |P(T)|, \tag{11}$$

how close the best of them lands with no optimizer involved (`pk_margin.py`). The 60 draws are a budget, not a property of the target, and the direction that matters is known: sampling more can only *lower* Δ_{bp}, never raise it, so a larger probe would move targets into the $\Delta_{\mathrm{bp}}{=}0$ stratum and never out of it. The 84% below is therefore a *lower* bound on how much of Δ_{single} sits where search is not required, and the residual $+0.016$ an *upper* bound on what the operator buys. Both errors run against us, which is why we did not spend more compute enlarging the probe.

It is a better predictor of our own failures than anything we optimized. Δ_{bp} separates the targets we solve under pKiss from those we do not at $\mathrm{AUC} = 0.824$, against 0.780 for length and 0.624 for the crossing-pair count, and it is not a restatement of that count ($r = +0.124$). The consequence for the headline rate is blunt: **the median of our 43 pKiss-solves has $\Delta_{\mathrm{bp}} = 0$.** Thirty of the 43 are hit *exactly* by at least one of 60 random compatible sequences. For most of what we count as solved, the operator is not what got us there.

Stratifying Δ_{single} by whether the probe can reach the target at all (Fig. 13): on the 30 targets with $\Delta_{\mathrm{bp}} = 0$ the panel operator takes 41 discordant wins to the single-oracle operator's 5 ($+0.080$); on the 30 with $\Delta_{\mathrm{bp}} > 0$ it takes 7 to 0 ($+0.016$, exact McNemar $p{=}0.016$). So 84% of the headline effect sits where a random sequence already succeeds. Something does survive where search is genuinely required, and we now report *that* as the size of the effect. This is the fourth independent measurement pointing the same way: undirected search matches guided search, the decomposition

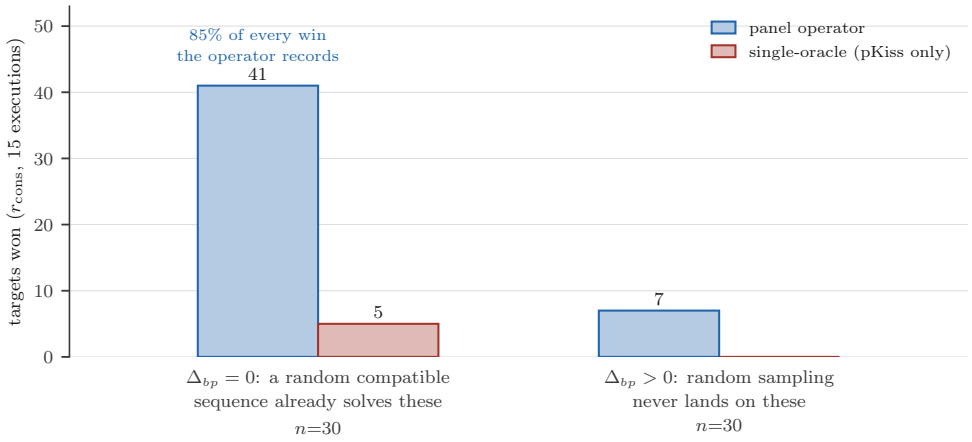

Figure 13: **The largest caveat this paper raises against its own positive result.** Discordant wins over the 15 executions, with the 60 targets split by whether any of 60 merely *compatible* random sequences already hits the target exactly. Almost the whole separation between the panel operator and the single-oracle operator sits on the half where random sampling has already succeeded, so on most of what we count as solved the operator is not what got us there. The residual on the other half (7 to 0, exact McNemar p=0.016) is what §E.4 reports as the size of the effect. Strata and counts parsed from `pk_margin.txt`.

scaffolding is inert, the objective ablation is unattributable, and now the separation is concentrated on zero-margin targets. The consistent reading is that the dimension we spent our design effort on is not the one that decides the outcome. What this does *not* touch is the oracle-gaming measurement: $43 \to 1$ and the paired $2/43$ versus $26/43$ carry-over are statements about adjudication, and whether a design survives a second predictor is a different question from whether its target was easy to hit.

Which pool is which, and a structurally disjoint evaluation. "Held-out" and "transfer" have been doing too much work, so the pools are named once (Table 15) and those names used throughout. Two facts belong in the open. The *primary* n=60 *pool is a development/evaluation pool*: the operator's design and every hyperparameter were fixed on it, so its rates are in-development performance. And the ID-disjoint pool [126:250] is disjoint by target *identity*, not by structure: $39/125$ of its targets are exact structural duplicates of a touched target and $94/125$ lie within 15% (App. I). So results there are an ID-held-out evaluation, not a transfer claim.

So we built the split the evidence calls for: cluster all 251 crossing targets by single linkage at 15% normalized dot-bracket distance and discard every cluster containing *any* target from ids 0–125. *The first thing this measures is the benchmark.* 251 targets $\to$ 60 clusters $\to$ 24 **clean ones holding** 27 **targets**, at a minimum realized distance of 0.167. Pseudobase++ is *structurally exhausted*. No clean transfer test on it can be well powered at a $\sim 5\%$ base rate, where the expected count is $\approx 1.4/27$. That is a property of the benchmark, not of the method, and we suspect it constrains other work reporting on it.

Within that limit the direction survives cleanly, and this is the strongest single piece of evidence for it. Over 5 seeds $\times$ 3 executions the panel operator clears the panel on these structurally novel targets and the single-oracle operator never does, at either stringency. Table 14 gives both. The result we lean on is the cell-level sign test, $15/15$ executions at both stringencies, and not the interval.

The operator's rate on structurally novel targets is *not lower* than on development (5.3%), and the search-free probe says this pool is measurably harder (median Δ_{bp} 0.176 against 0.059; 15% of targets reachable by a random compatible sequence against 50%). Scoring higher on a harder pool is the opposite of the memorization signature a structural split is built to detect. We deliberately do *not* lean on the interval ($[+0.030, +0.116]$ at n=27), which is underpowered by construction; and the sign test licenses "a fresh execution on *these* structurally novel targets", not "on a fresh structurally novel target set". The honest summary: Δ_{single}'s *sign* now has a structurally disjoint replication, its *magnitude* remains unestimable on any pool this benchmark can supply.

Table 14: **The structure-aware transfer test, at both stringencies** (`sdis_stats.py`): 27 targets in 24 clusters, each at minimum single-linkage distance 0.167 from every target used in development, run at 5 seeds × 3 executions. The comparison is paired within cell. b counts cells where the panel operator succeeds and the single-oracle operator does not, c the reverse; c is 0 under the strict rate, so there is no execution in which single-oracle optimization reaches a structurally novel target that panel optimization misses. *The interval is the column not to lean on*: at n=27 with a ∼5% base rate this pool is underpowered by construction, which is a finding about Pseudobase++ rather than about either arm (§E.4). The exact McNemar assumes 405 independent cells drawn from 27 targets and is printed only to show the correction; the sign test over executions is what we quote. Note the direction of the last row: the operator's strict rate on structurally novel targets (6.9%) is *above* its development-pool rate (5.3%), on a pool the search-free probe measures as harder.

stringency	**single-oracle** mean / 27	**panel op.** mean / 27	**paired Δ**	b / c	**sign test** over 15 cells	**bootstrap** 95% CI
r_{in} (pKiss ∧ ProbKnot)	0.27	8.60	+0.309	127 / 2	15/15, p=6.1×10^{-5}	[+0.188, +0.435]
r_{cons} (all three predictors)	**0.00**	1.87	+0.069	28 / **0**	15/15, p=6.1×10^{-5}	[+0.030, +0.116]
anticonservative, shown for contrast only: exact McNemar p=2.5×10^{-35} (r_{in}) and 7.5×10^{-9} (r_{cons})						

E.5 The Panel's Recall Ceiling, and the One Unbounded Axis

The panel is a low-recall instrument, and that bounds every number above. On 80 native bpRNA pseudoknots, folding each native's *own* sequence, per-predictor crossing recall is pKiss 20%, ProbKnot 4%, ShapeKnots 11%, and the three-way conjunction **1**% (1/80). Two consequences we carry throughout: a low r_{cons} is weak evidence of *absence* of capability, and r_{cons} is a high-stringency consistency measure, not a lower bound on true capability. The predictors also have *no fixed stringency order*: on natives it is pKiss > ShapeKnots > ProbKnot and on designs it inverts, because agreement measured on designs is dominated by model-family sharing between the predictor and the objective that produced the design.

What can an instrument this insensitive resolve? Two things must be kept apart. *Resolution* is a property of n, the pairing and the base rate; recall does not enter. Simulating the design actually run (60 targets × 5 seeds, target-clustered bootstrap, base rate 4/60; `resolution.py`) gives 80% power at an observed gap of ≈ 6/60, so it separates 4/60 from ≈ 8/60 and 5/60 from 15/60 at power 1.00, but not 4/60 from 5/60. That places our own headline: Δ_{single} at +0.067 sits at 77% power, at the design's resolution limit, which is "established, narrowly" in units a reader can check. *Attenuation* is where recall enters: an observed gap d stands for a true gap d/ρ, so low recall rescales the comparison rather than blurring it and makes every rate conservative. The rescaling assumes ρ is common to both arms, which holds for r_{cons} and not for carry-over whose point is that ρ differs by arm; we use it only for the former.

Generalization, and its limit. On a second disjoint Pseudobase pool the regime repeats (0/50 floor, 5/50 panel-unanimous); on a cross-source bpRNA pool our operator *and* the SOTA solver both reach 0/60. The margin probe says most of that is the pool: its median Δ_{bp} is 0.581 against the development pool's 0.059, and 5% of its targets are reachable by a random compatible sequence against 50% there. The negative side generalizes across datasets; the positive claim is limited to the Pseudobase++ distribution. Statistical conventions, the leave-one-target-out check, the quasi-held-out/exploratory split and the per-oracle cross-source breakdown are in App. G and App. H.1.

The one axis that is not structurally exhausted, run at scale. Every magnitude above is unestimable on Pseudobase++, and the resolution calculation says why in units: the design reaches 80% power at an observed gap of ≈6/60, and Δ_{single} is 4/60. The parameterized family $\mathcal{L}_{\text{H}} = \{T_{n,m}\}$ has no such ceiling. It is non-context-free for every $m \geq 1$, it is the family the class-level reading is about, and it is unbounded, so power there is a budget decision rather than a property of a benchmark. An earlier revision ran 16 of its members at one seed and filed them in an appendix as not relied on; that was the wrong call. We run the full grid $n, m \in \{3, \ldots, 12\}$: 100 targets, five seeds, deterministic, 1500 cells (`family_grid.py`, `family_stats.py`).

Table 15: **The pools, named by their actual contact with development.** We use these names throughout and retire the earlier label "strictly untouched", which invited a structural reading the index split does not support. The last row is the structure-aware split the evidence calls for; we built and ran it, and its size is a finding about the benchmark. The Δ_{bp} column is the pool's median search-free margin (Eq. (11), 60 random compatible sequences per target): the pools are *not* difficulty-matched, and two readings in this paper turn on that (`runs/cc_ascent/pool_margin.py`).

name used here	**ids /** n	**med.** Δ_{bp}	**contact with development, and what it can support**
development / evaluation pool	0–59 (60)	0.059	operator design and *all* hyperparameters fixed here. Supports in-development comparison between arms; **not** a held-out benchmark, and we do not call it one. Half of it ($30/60$) has $\Delta_{\mathrm{bp}}{=}0$ (Eq. (11)): reachable by a random compatible sequence, and carrying 84% of Δ_{single}
inspected secondary pool	60–109 (50)	—	inspected during development but not tuned on. Supports a same-regime replication check
invention training pool	110–125 (16)	—	the agent-invention loop's training set (App. J)
ID-disjoint follow-up pool	126–250 (125)	0.000	untouched by development, tuning and model feedback, but *structurally overlapping* with the touched set ($39/125$ exact duplicates, $94/125$ within 15%; App. I). Supports an *ID-held-out evaluation*; does **not** support a transfer claim
cross-source pool	bpRNA (60)	$\mathbf{0.581}$	independent source; both our operator and the SOTA solver score $0/60$, on a pool an order of magnitude harder than development on the search-free coordinate, which is most of why
structurally disjoint pool	24 clusters (27)	$\mathbf{0.176}$	whole structure clusters assigned, keeping only clusters with *no* touched member; every target $> 15\%$ from every developed-on target. Supports the *direction* of Δ_{single} ($15/15$ executions, single-oracle $0/27$ throughout; §E); too small for its magnitude, and that smallness is a fact about Pseudobase++, whose 251 targets contain only 60 structural clusters. *Harder* than development (15% of its targets reachable by a random compatible sequence against 50%), which is why the panel operator's higher rate there counts for more, not less

The regime reproduces and sharpens. The CF floor is $\mathbf{0/300}$, so the negative side's empirical counterpart holds at five times the pool size. The single-oracle operator reaches $500/500$ under the oracle it optimizes and $9/500$ under three, on a set where every target is non-CF by construction. The panel operator reaches $167/500$. Paired within (target, seed) and clustered on target, $\Delta_{\mathrm{single}} = +0.316$ $[+0.258, +0.378]$, $b{=}159$ against $c{=}1$, exact McNemar $p{=}2.2{\times}10^{-46}$. **That is the first estimable magnitude in this paper**, five times the Pseudobase++ estimate, which is a statement about the two target sets and not about the operator.

The success-vs-size curve, and the diagnostic that stops us mis-reading it. Panel-unanimity rises with the family parameter, from 0.186 at $n{+}m \leq 12$ to 0.529 at $n{+}m \geq 18$. That reads as a fixed-description operator holding up as the family grows, and Fig. 14 refutes it: the operator's own success rate *falls* with size ($0.900 \rightarrow 0.736$) while carry-over to the held-out predictor rises sharply ($0.206 \rightarrow 0.718$). What grows is not the procedure's reach but the panel's willingness to confirm. The honest reading is narrower and still useful: there is no size cliff in the confirmed rate up to $n{+}m{=}24$, which is what a uniform procedure needs and a memorized table cannot supply, but the operator does have a size-related limitation and it is visible in r_{in}. The single-oracle operator's pKiss→ProbKnot agreement moves the *other* way with size ($0.093 \rightarrow 0.036$), which rules out a blanket size effect on predictor agreement and is the clearest support in this paper for the success-predicate reading.

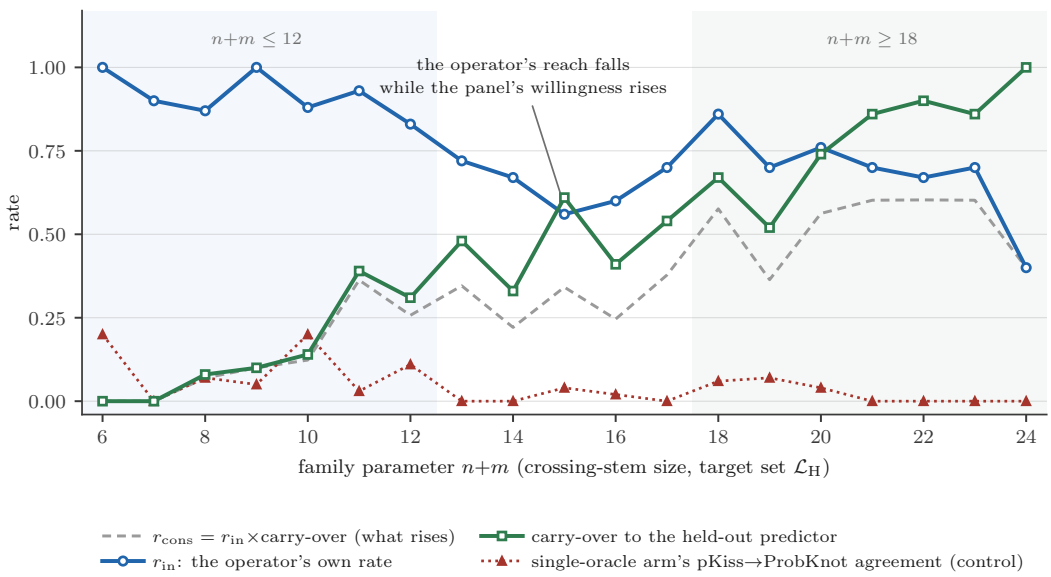

Figure 14: **Why the rising confirmed rate on $\mathcal{L}_{\mathrm{H}}$ does not mean what it looks like.** The panel-unanimous rate r_{cons} (grey) rises with the family parameter, which reads as a fixed-description operator holding up as the family grows. It is a product, $r_{\mathrm{cons}} = r_{\mathrm{in}} \times$ carry-over, and only the second factor rises: the operator's own reach *falls* between the shaded strata while carry-over to the held-out predictor more than triples, because longer crossing stems are thermodynamically dominant enough that a predictor which never saw them agrees anyway. The single-oracle operator's pKiss→ProbKnot agreement (red) is the control that blocks the obvious objection, since it moves the *other* way: "longer stems make predictors agree" is not a blanket effect, and satisfying two predictors on a long stem selects sequences whose crossing stem dominates where satisfying one does not. 100 targets $\times$ 5 seeds, deterministic (`family_grid.py`, `family_stats.py`).

Scope, stated beside the numbers rather than under them. These rates are not comparable to the Pseudobase++ rates and must not be quoted alongside them. This is one synthetic distribution, with a uniform topology and no natural sequence constraints. It is also the exact topology the operator's decompose–compose scaffold was built for — a home-field advantage we name rather than discount. And small-size failures may be undesignability, since we ran no designability reference on this grid. What the family buys is the one thing Pseudobase++ cannot supply at any n: a magnitude, on targets sharing no structural cluster with anything we developed on. It buys no generalization.

Where this leaves the ascent. An operator optimizing panel agreement beats the single-oracle operator's $1/60$, if modestly: 3–$8/60$ across five seeds, 1–$9/60$ over all 24 executions, and 2–$17/60$ for agent-invented operators (App. J), above a decidable $0/60$ floor. These are mixed-panel-consistent, autonomous crossings on the Pseudobase++ distribution; we do not claim they are the first in the literature, only the first we observe under this protocol. On those designs the target is the pKiss MFE for all 17 whose suboptimal enumeration completed, dominant by a median 1.2 kcal/mol, so the positive verdict is not a marginal coincidence of pair sets, within the pKiss energy model, and on the shorter designs only. The remaining gap ($\sim 5/60$) is itself the finding: robust autonomous pseudoknot design is hard, and two-oracle agreement generalizes to a third only partially.

F PROOFS

Proof of Proposition 2. The pseudoknot-free oracle Ω_{cf} returns a minimum-free-energy secondary structure in the pseudoknot-free ensemble; its representation admits no crossing pair, so $\Omega_{\mathrm{cf}}(x) \in \mathcal{L}_{\mathrm{nest}}$ for every sequence x. Hence for any policy π, every $T \in \mathcal{R}_{\Omega_{\mathrm{cf}}}(\pi)$ satisfies $T = \Omega_{\mathrm{cf}}(\pi(T)) \in \mathcal{L}_{\mathrm{nest}}$, i.e. $\mathcal{R}_{\Omega_{\mathrm{cf}}}(\pi) \subseteq \mathcal{L}_{\mathrm{nest}}$. If $T \in \mathcal{L}_{\mathrm{pk}}$ then T contains a crossing pair, so $T \notin \mathcal{L}_{\mathrm{nest}}$ and therefore

$T \notin \mathcal{R}_{\Omega_{\mathrm{cf}}}(\pi)$. Testing $T \in \mathcal{L}_{\mathrm{pk}}$ (a crossing pair among $|P|$ pairs) is $O(|P|^2)$ and independent of any search over sequences; the non-membership is thus decided offline. □

Decision procedure for the negative side (and its complexity). Given T with pair set $P(T)$, the crossing test returns CROSSING iff $\exists\,(i,j),(k,l) \in P(T)$ with $i < k < j < l$. The naive double loop is $O(|P|^2)$; sorting $P(T)$ by left endpoint and sweeping with a stack of open right-endpoints (pushing j at i, popping at j, flagging a violation if a new pair opens inside an open interval but closes outside it) decides it in $O(|P| \log |P|)$. The verdict for the entire held-out pool is computed *once, offline*, before any sequence is designed. No folding, search, or evaluation over Σ^* is involved. This is the precise sense in which the negative side is a theorem discharged before any run (contrast the positive side, which requires folding each committed design). **Class boundary (invoked), and the grammar witness (worked).** That $\mathcal{L}_{\mathrm{nest}}$ is context-free, and that the crossing families we treat are non-context-free yet admit MCFG grammars of dimension ≤ 2, are established (Rivas & Eddy, 2000; Nebel & Weinberg, 2012; Kato et al., 2006); we do not reproduce their proofs, and we make no claim that every pseudoknot topology is captured at dimension 2. We do make the witness explicit for the family the operator targets.

The parameterized crossing family. Let

$$\mathcal{L}_{\mathrm{H}} = \left\{ \; (^{\,n}\, \ell\, [^{\,m}\, \ell\,)^{\,n}\, \ell\,]^{\,m} \;:\; n, m \geq 1 \right\}, \tag{12}$$

the canonical H-type pseudoknots (ℓ a fixed unpaired linker): a nested round stem of n pairs crossed by a square stem of m pairs. Its pairing skeleton is the interleaving $\{a^n b^m c^n d^m\}$ under $a{=}($, $b{=}[$, $c{=})$, $d{=}]$.

Non-context-freeness, and what is invoked versus verified here. The general results are invoked, not re-proved: that crossing arcs are not context-free is Nebel & Weinberg (2012), and that pseudoknot classes sit in low-dimension MCFG subclasses is Kato et al. (2006). What we do verify is the witness for *our* family, since we constructed it and no citation covers it: $\mathcal{L}_{\mathrm{H}}$ carries the two equalities $\#a{=}\#c$ and $\#b{=}\#d$ of the count language, and no $uvwxy$ decomposition with $|vwx| \leq p$ can span more than two of four blocks, so pumping breaks one equality (Ogden, 1968; Hopcroft & Ullman, 1979); hence no single-stack generator realizes the family uniformly. This matters for one thing only, and it is the thing §E's grid rests on: it is what licenses calling a synthetic ladder a test of the same capability rather than an unrelated construction.

An MCFG of dimension 2 that does. Two dimension-2 nonterminals each emit a *pair* of substrings (the two ends of one stem), and the start rule interleaves them:

$$\begin{aligned} R(a\,x_1,\; c\,x_2) &\leftarrow R(x_1, x_2); & R(\varepsilon,\varepsilon) &\leftarrow \ \text{(base)};\\ Q(b\,y_1,\; d\,y_2) &\leftarrow Q(y_1, y_2); & Q(\varepsilon,\varepsilon) &\leftarrow \ \text{(base)};\\ S(x_1\,\ell\,y_1\,\ell\,x_2\,\ell\,y_2) &\leftarrow R(x_1, x_2),\; Q(y_1, y_2). \end{aligned} \tag{13}$$

R derives the round-stem pair (a^n, c^n), Q the square-stem pair (b^m, d^m), and S places them at four interleaved slots, exactly the crossing a single yield cannot produce. Dimension 2 (each nonterminal coordinates two substrings) is sufficient (the grammar above) and necessary for this family (dimension 1 is exactly CF, excluded by the pumping argument), i.e. the minimal machinery above CF (Seki et al., 1991; Kato et al., 2006).

Definition, result, and inference, kept apart. The sentence "realizing $\mathcal{L}_{\mathrm{H}}$ uniformly is an MCF capability" is a **definition**, and it quantifies over *all* (n, m): a finite-description operator that, for *every* (n, m), returns a sequence whose fold is S's yield would stand in for the grammar itself, which no CF generator can. Our **experimental result** is strictly weaker and finite: the frozen operator succeeds on the tested ladder $n{=}m \in \{5, \ldots, 14\}$ and selected asymmetric (n, m) up to $(14, 3)$, at parameters 3–4× beyond those it was fixed on (§E). The **inference** between them is therefore an extrapolation, and we label it as such: the result is *consistent with* uniform coverage of the family and rules out a lookup table of the development targets, but it does not establish the definition's universally quantified claim. A finite program with length-parameterized templates would pass the same ladder without realizing every member, and the three asymmetric-extreme failures show the operator is not in fact uniform at the tested budget. We never write that the operator realizes $\mathcal{L}_{\mathrm{H}}$ uniformly; we write that it transfers to tested members well beyond its development range.

Worked derivation (n=m=2, linker ℓ=`....`). Eq. (13) derives $T_{2,2}$ = `((....[[....))....]]` in four steps: two applications each build the stem pairs (aa, cc) and (bb, dd), then the start rule interleaves them across the four slots separated by ℓ. A CFG cannot reproduce this table: its single-yield nonterminals cannot carry the two ends of a stem to separated, interleaved positions.

The pseudoknot classes form an MCFG-dimension hierarchy. The dimension-2 grammar above captures the H-type family (one crossing stem). Richer pseudoknots (kissing-hairpin interactions, complex $\geq$ 3-stem entanglements) are widely expected to require higher MCFG fan-out, but the minimal fan-out for each specific class, and the inclusion/incomparability relations among pseudoknot classes as formal languages, are delicate questions we do not settle here (Rivas & Eddy, 2000; Nebel & Weinberg, 2012; Kato et al., 2006). We therefore assert the dimension-2 witness *only* for the H-type family we design against and make *no* per-class dimension-hierarchy claim: in particular we do not claim that MCFG dimension grows with the crossing-stem count. The grades order targets by a structural difficulty statistic (crossing-stem count), so the empirical frontier organizes by grade, without asserting that statistic is a formal grammar grade. The CF $\rightarrow$ MCF boundary, the object of this paper, is crossed already at the first tier (one crossing stem); higher tiers are a finer hierarchy *within* MCF that we do not attempt to climb.

Table 16: The oracles as maps $\Sigma^* \rightarrow \mathcal{T}$: folding model, range, and pseudoknot subclass modeled.

oracle	folding model	range $\operatorname{im}\Omega$	subclass
Ω_{cf} (ViennaRNA)	Turner-2004 MFE, pseudoknot-free ensemble	$\subseteq \mathcal{L}_{\mathrm{nest}}$	none (nested)
pKiss	heuristic min-energy folding with knot motifs (Janssen & Giegerich, 2015)	$\subseteq \mathcal{T}$	H-type, kissing hairpin
ProbKnot	max-expected-accuracy pairs from a partition function (Bellaousov & Mathews, 2010)	$\subseteq \mathcal{T}$	pairs above a probability cut
ShapeKnots	MFE under SHAPE restraints (here run *de novo*) (Hajdin et al., 2013)	$\subseteq \mathcal{T}$	low-order pseudoknots

The folding oracles (models and output spaces). Table 16 fixes each oracle as a map $\Sigma^* \rightarrow \mathcal{T}$ with an explicit range. $\operatorname{im}\Omega_{\mathrm{cf}} \subseteq \mathcal{L}_{\mathrm{nest}}$ (generally strict, as undesignable nested structures are unrealizable) is exactly what Prop. 2 (via Lemma 1) requires; the three panel predictors admit crossing outputs but model *different* pseudoknot subclasses under different objectives, which is the mechanistic source of the $[15, 60]\%$ cross-oracle envelope of §C.1 and hence of the abstention rule.

G Reproducibility and Protocol Details

Oracles are computed under a fixed configuration. The submission supplement provides a bounded verdict-level re-analysis release: reduced per-target records and two offline statistical scripts, not all sequence-level records, design code, or intervention implementations. The CF ladder is ViennaRNA (Turner 2004, 37°C); the pseudoknot panel is pKiss (reference) with ProbKnot and ShapeKnots for the oracle-robustness envelope and abstention. The sealed-commitment step records a hash and timestamp of each prediction before the held-out adjudicator is run; the $\pm$solver ablation toggles the external pseudoknot solver while holding the agent fixed; the minimal-hint sweep (E3), a planned minimal-hint probe, parameterizes how much of the target's crossing structure is revealed and reports the smallest hint under which the operator crosses. Every measured number in this manuscript enters only from such a logged run (§E). The supplementary README distinguishes included verdict-level re-analysis from end-to-end replay, which this archive does not support.

Software versions, commands, and release. The oracles are pinned: ViennaRNA 2.7.2 (Turner 2004 parameters, 37°C) for the CF ladder and inverse folding; pKiss 2.3.0 (Bellman's GAP compiler `bellmans-gapc` 2024.01.12, bioconda) as the reference pseudoknot oracle; RNAstructure 6.6 for ProbKnot and ShapeKnots (the latter run *de novo*, no SHAPE constraints); all under Python 3.10. Each oracle runs in its own pinned conda environment; the harness records the exact command line and the raw predictor output for every call. Design seeds are the integers 0–4.

Execution-level nondeterminism at a fixed seed, and the fix. Two independent runs of the same arm at the same nominal seed gave different counts. Our first diagnosis, subprocess timing under a wall-clock cap, was incomplete: the operators also call ViennaRNA's `inverse_fold`, a stochastic search on ViennaRNA's *own* global RNG, which is invisible to `random.Random(seed)` and advances from call to call. The operators now expose `deterministic=True`, which replaces both wall-clock caps with counted ones (iterations plus a 150-call pKiss allowance, above the 142.7 the hungriest arm realized under the clock) and seeds ViennaRNA's generator per attempt. Validated against a 96-process synthetic load: $4/4$ and $4/4$ targets bit-identical for the two operators against $1/4$ and $0/4$ with the default. Independently confirmed by `family_grid_par.py`, which re-runs a serially-produced seed under 50-way parallelism and finds per-target verdicts identical, so a (target, seed) pair fixes the run regardless of what else is executing.

How large the noise was, and why the paired reading survived it. Pooling all six independent executions per nominal seed (`exec_spread.py`), the panel operator's r_{cons} has a pooled within-seed s.d. of $\mathbf{2.04/60}$, the *same order* as the execution-replicated method effect of $2.87/60$: enough to make an unpaired comparison of two separately-reported counts meaningless, not enough to swamp the paired one (4.3 standard errors from zero). The single-oracle operator is nearly deterministic (0–1, mean spread $0.3/60$), so pairing rescues the comparison though neither count is individually stable. Two mechanisms drove the spread: subprocess timing jitter, and *sweep composition*, since the ablation sweep ran six arms per worker and the paired sweep two, so the same nominal configuration got through 87.3 versus 130.7 pKiss calls per target. A wall-clock-budgeted operator's result is thus a function of what else was running beside it, which no seed records. Four consequences we state rather than hide. Per-seed counts in Table 9 are single-execution draws, to be read with at least $\pm 2/60$ granularity. A bootstrap over targets and seeds only understates uncertainty by one level, so the wall-clock intervals quoted in §E use a three-level bootstrap. Every such comparison is paired *within* an execution, so this variation inflates intervals but cannot masquerade as a method effect. And no claim should be read off a single count.

CF floor measurement. The floor is the CF-confined REMC optimizer run on the same targets under the same adjudicator, at the same nominal caps as the audited arms. It scores $0/60$ on crossing targets, which is an empirical result at a finite budget and not a restatement of Prop. 2: the proposition bounds what the pseudoknot-free *verifier* can certify, the floor records what the prior *policy* actually produced.

The three tests behind Table 9. Let $u_T^{(A)} \in \{0,1\}$ indicate whether arm A realizes held-out target T panel-unanimously, over a pool of size N. *Percentile bootstrap*: a 95% CI for a rate resamples the N targets with replacement B=20,000 times. *Paired McNemar* (same targets, arms A vs. single-oracle G): with $b = \#\{T : u_T^{(A)}{=}1, u_T^{(G)}{=}0\}$ and $c = \#\{T : u_T^{(A)}{=}0, u_T^{(G)}{=}1\}$,

$$p_{\text{McN}} = \min\Big\{1,\ 2\sum_{k=0}^{\min(b,c)} \binom{b+c}{k} 2^{-(b+c)}\Big\}. \tag{14}$$

Fixed-baseline binomial: against the single-oracle rate $p_0{=}1/60$, the one-sided p-value for k successes in N trials is $\sum_{j=k}^{N} \binom{N}{j} p_0^j (1-p_0)^{N-j}$. The analysis is `stats_separation.py`, run offline on the saved per-target adjudications.

Statistical conventions. Rates use a fixed denominator of the full pool: a target on which the three predictors disagree counts as not-solved, so abstention withholds a positive claim rather than dropping the target. Every comparison carries an effect size, never a p-value alone. Three levels of variation matter (target, nominal seed, execution at a fixed seed) and are not interchangeable, so we treat the *deepest* interval run for a comparison as primary and label every interval with what was resampled. With five nominal seeds the level-1 resampling distribution has few distinct states, so these are *sensitivity intervals* with no coverage claim, which is why the most robust statement about Δ_{single} uses no interval at all: the operator is strictly higher in 14 of 15 independent executions, and 15 of 15 on the structurally disjoint pool.

Is the effect carried by a few lucky targets? r_{cons} is a low-base-rate conjunction, so we checked. Deleting each target in turn moves the paired Δ within $[+0.068, +0.081]$ around $+0.080$; the most

influential single target moves it by -0.012. The successes do concentrate, 25 target-seed wins on 13 distinct targets with the top three accounting for 44%, which we report instead of treating as 25 independent wins; but they are not one or two targets, and their mean crossing-pair count (5.46) matches the pool (5.48). Targets are also not independent draws: single-linkage clustering at 15% collapses the 60 into 18 clusters, one holding 29, so we resample clusters, which widens the interval from $[+0.040, +0.127]$ to $[+0.024, +0.149]$ and still clears zero.

Quasi-held-out versus exploratory, stated once. We do not use the word "confirmatory": we hold no timestamped pre-registration, and a confirmatory held-out analysis would need the evaluation pool disjoint in the respect that matters for a structure-conditioned operator, namely structure. *Quasi-held-out*: the paired r_{cons} separation on the 125 ID-disjoint targets. *Exploratory*: the ShapeKnots-only comparison, the crossing-pair-count association, the budget sweep, the six-model invention draws and the cross-source pool. No family-wise correction is applied, since these answer different questions on overlapping data; no exploratory p-value here is a test, and "significant in $k/3$ seeds" is a descriptive replication count.

The multi-level bootstrap, specified in full. *Estimand*: Δ, the mean over (target, seed, execution) cells of the paired indicator difference between two arms. *Pairing*: the unit is the paired difference formed *inside* one cell, so both arms carry identical resampling weights at every level. *Level 1*: draw nominal seeds with replacement. *Level 2*: executions are nested within seed; we report *full-nest* (three per drawn seed, matching the design run, primary) and *one-exec* (one per drawn seed, wider, a conservative sensitivity). *Level 3*: draw targets with replacement, *synchronously* for every drawn cell and both arms, reported with the target as the unit and with the structural cluster as the unit (single linkage at 15% normalized dot-bracket distance), since the pool clusters (App. I). *Interval*: percentile, 2.5/97.5 quantiles of B=20,000 replicates. *Why no coverage claim*: with three nominal seeds the level-1 resampling distribution has few distinct states, so these are *sensitivity intervals* and we do not use them as tests. *Implementation*: Python `random` seeded at 20260726 in `pabrep_stats.py`, `gvp_stats.py` and `cluster_boot.py`.

Is the index split a *structural* split? (An audit, with a partly negative answer.) Splitting by target index guarantees no target is reused, but it does not guarantee the untouched targets are structurally distant from the touched ones, and if they were near-duplicates, "held-out transfer" would reduce to template-neighbour generalization. We measured it (`runs/cc_ascent/split_audit.py`): for each of the 125 ID-disjoint targets (ids 126–250) we found its nearest neighbour among the 126 touched targets (development [0:60], inspected pool [60:110], agent-invention training [110:126]) under normalized dot-bracket edit distance. The result is a genuine caveat: the nearest-neighbour distance has median 0.077, and $39/125$ **ID-disjoint targets are *exact* structural duplicates** of a touched target, with $94/125$ within 15%. So the index split is *not* a structural split, and it cannot carry a transfer claim (Table 15).

Two observations that bound the problem without disposing of it. First, structural duplication is not sequence homology here: the IUPAC restraint strings of those nearest-neighbour pairs still differ by a median 0.329 normalized edit distance (minimum 0.243), so a duplicated *target* does not come with a duplicated *sequence* to copy. We are careful about what this does and does not exclude. The operator's input is the target *structure*, so a differing natural sequence rules out only the narrowest leakage channel, copying a neighbour's sequence, and leaves untouched every channel that runs through the target: objective design, mutation heuristics, template construction, the stopping criterion, error weighting, and researcher-level choices all saw dot-bracket strings that recur in the ID-disjoint pool. "Different sequence" is therefore not evidence against template leakage.

Second, success does not visibly concentrate on the duplicates, but the sample cannot establish that it does not (`runs/cc_ascent/cluster_boot.py`). Pooled over three seeds, the panel operator's r_{cons} is $2/117$ (0.017) on the 39 *exact* duplicates, $19/165$ (0.115) on the 55 near-but-not-exact neighbours, and $9/93$ (0.097) on the 31 structurally distant targets. So if anything the exact duplicates are the *worst* stratum, the opposite of the memorization signature. Taking near and exact together against distant, and making the *structural cluster* the resampling unit inside each stratum (18 clusters among the 94 near-or-exact targets, 28 among the 31 distant ones), $\Delta_{\text{duplicate}} = p_{\text{distant}} - p_{\text{near}} = +0.022$ with a cluster bootstrap 95% interval of $[-0.048, +0.108]$. That interval excludes neither a substantial positive nor a substantial negative difference, and the strata are not difficulty-matched by design (mean crossing-pair counts 5.41, 5.35, 5.55) though they happen to be

close. The correct statement is therefore the narrow one: *we did not observe higher success on the near-duplicate subset, but the structurally non-disjoint split prevents a clean transfer claim*, and not that the failure mode is absent. This pool therefore supports an ID-held-out evaluation only.

The remedy, run. The right design assigns whole structure clusters rather than individual targets, and §E reports it: clustering all 251 targets at the same 15% threshold and keeping only clusters with no touched member yields 24 clusters holding 27 targets, verified at minimum distance 0.167 from the touched set. Two results. First, about the *benchmark*: 251 crossing targets contain only 60 structural clusters, of which 36 touch our development, so Pseudobase++ cannot furnish a well-powered structurally clean transfer test for an effect at a $\sim 5\%$ base rate, a constraint we suspect applies to other work reporting on this benchmark and which we therefore state plainly. Second, about the *effect*: on that pool the panel operator is mixed-panel-consistent on $1.87/27$ (6.9%, i.e. *above* its 5.3% development-pool rate) against the single-oracle operator's $0.00/27$ in all fifteen executions, $b{=}28$ against $c{=}0$. The near-duplicate concern the audit raised is thus tested directly and does not materialize. But on 27 targets that settles direction, not size.

H Robustness Checks, the Adjudicator's Reference Class, and Materials

H.1 Robustness checks in full

§E states the two checks the argument uses. This appendix adds the three details it omits.

The skeleton negative control, and why we do not call it specificity. Folding each target's CF skeleton under the panel, the panel credits the crossing on $0/60$ skeletons while recovering the nested skeleton on $57/60$, whereas a naive A_{nest} evaluator over-accepts them. Skeletons are one narrow negative class, and a real negative distribution would also contain near-miss designs, mispaired variants and *alternative* pseudoknot topologies on which the panel is untested. The defensible statement is exactly *no observed false positives on the skeleton negative control*; the false-positive rate is uncalibrated, so we treat r_{cons} as a high-stringency consistency measure and not a bound.

The cross-source $0/60$, per oracle. It does not let us blame the strictest predictor alone: DesiRNA forms $9/60$ crossings under pKiss but only $3/60$ survive ProbKnot and $0/60$ pass ShapeKnots, so ProbKnot and not just de-novo ShapeKnots already rejects most, for the SOTA solver as much as for ours (our operator reaches 1–$2/60$ under pKiss $\wedge$ ProbKnot across two runs of the same arm, and $0/60$ either way). A larger repair budget (240 s, 2500 iterations) does not change it.

Budget, which established nothing. Over $\{60, 180, 300\}$ s on a fixed 20-target subset, panel-unanimous is $\{1, 3, 1\}/20$: no clean budget trend, dominated by search variance, consistent with oracle-guided search rather than a budget-monotone capability.

Growing-parameter extrapolation: the earlier 16-target version. *Superseded by the 100-target grid of §E, and kept only so the record shows what was run first.* A finite target set cannot establish a class ascent, so we tested the strictly weaker property a finite set cannot exhibit at all: freeze the operator on small stems ($n{=}m \leq 4$) and apply it unchanged to a larger unseen ladder of $\mathcal{L}_{\mathrm{H}}$ (to $n{=}m{=}14$, length 68 nt). The CF floor is $0/16$ throughout and the frozen operator realizes $13/16$ under pKiss $\wedge$ ProbKnot including the whole diagonal $n{=}m{=}5 \ldots 14$, but strict three-oracle unanimity is $6/16$, and a finite program with length-parameterized templates could produce exactly this. The result is therefore consistent with unbounded-family coverage and equally consistent with a template, does not distinguish an oracle-agreement ceiling from a search-budget limit, and no claim in this paper rests on it. We record it so that its absence from the main text is not selective reporting.

H.2 Adjudicator choice, abstention scoring, and what ShapeKnots costs

The choice of held-out adjudicator deserves scrutiny because the whole positive side rests on it. ShapeKnots is the only crossing-aware predictor in the panel that neither operator optimizes, so it is the only one for which "held out" is true by construction rather than by convention. It also runs here in an *unvalidated de-novo mode*: the method was designed to fold under experimental SHAPE

constraints (Hajdin et al., 2013) and we supply none, so we are using it outside the regime it was calibrated for and say so wherever its numbers appear.

What that costs is stringency we cannot separate from strictness. Given its own reference class the judge is not implausibly harsh: it recovers 11.3% $[5.3, 20.3]$ of native pseudoknots from their own sequences (`heldout_ceiling.py`) and confirms our panel operator's designs at 8.0%, about 70% of that ceiling, against the single-oracle operator's 3.7%. The low absolute rates are therefore substantially a property of the adjudicator, and the *ratio* between arms is the part we read.

Three caveats keep that an orientation rather than a corrected estimate. The pooled interval ignores that the same 125 targets recur across seeds, so it is **too narrow**; the paired seed-level test is unchanged and still separates in only one of three seeds; and natives and designs are different populations, ours being $372/375$ H-type against $75/80$ natives carrying more than five crossing pairs. A related line of work found precisely the unstratified version of this comparison to be confounded by topology, and our native set carries no H-type/kissing label and only five low-crossing members, so we cannot stratify it. The ceiling-relative reading therefore changes what r_{ho}'s absolute size means; it does not change its verdict.

Abstention scoring. A target on which the three predictors disagree counts as not-solved against the full-60 denominator, so abstention withholds a positive claim rather than dropping the target, and every reported rate is conservative in the same direction. That is why r_{cons} is a high-stringency consistency measure rather than a selective-prediction rate: we never report an accuracy conditioned on the panel having agreed.

I Datasets, Intrinsic Grades, and Materials

Every target's difficulty *grade* is a property of the target fixed before design: its *formal-family type* (nested vs. crossing) and, within crossing, its *crossing-pair count* g, the number of base pairs that cross another, computed in $O(|P|^2)$ from the dot-bracket. Table 17 lists the pools. The cross-

Table 17: Datasets. Grade g = crossing-pair count. EteRNA100 is the nested (CF) competence baseline; the three crossing pools are the crossing capability sets; see Table 15 for what each supports.

pool	source	n / length	grade g distribution
held-out (primary)	Pseudobase++ (RNAInvBench) idx 0–59	60 / 21–137 nt	$g{:}n$ — 3:4, 4:6, 5:22, 6:18, 7:7, 8:1, 9:2
second pool	Pseudobase++ idx 60–109	50 / 21–137 nt	disjoint from primary
cross-source	RNAInvBench *inverse_rna_folding* (bpRNA/ArchiveII)	60 / 21–120 nt	$g{:}n$ — 1:4, 4:13, 5:6, 6:16, 7:8, 8:5, 9:3, 10:3, 12:1, 13:1
family $\mathcal{L}_{\mathrm{H}}$	synthetic H-type (Eq. (12))	10 / 24–44 nt	$n{=}m \in \{3, \ldots, 8\}$ + asym
EteRNA100-V2	(Koodli et al., 2021)	100 / 12–400 nt	nested (CF)

source pool is reconstructed to crossing dot-bracket from the pair list (crossing pairs are encoded as non-type-0; a greedy multi-bracket assignment yields ≤ 4 bracket levels); multiplet structures are excluded so each base is in ≤ 1 pair (`runs/cc_ascent/bprna_pk.py`). The two pools overlap on the crossing-pair count, so the cross-source drop (§E) is not explained by that one covariate. *We do not call the comparison "grade-fair"*: overlap on a single prespecified structural covariate is not distribution matching. The two pools are *not* matched on length, pseudoknot subtype, loop-length distribution, base-pair composition, motif family, source database, or sequence homology, any of which could drive the drop. The cross-source result should therefore be read as a distribution-shift observation with the shift only partly characterized.

J The Operator-Invention Protocol

An LLM agent (GLM-5.2, temperature 0.4, no output cap) is given a documented *primitive API* and asked to write one function `design(target,seed)`; it may not import anything or call an external pseudoknot solver. The primitives are `fold_cf` (ViennaRNA MFE, nested),

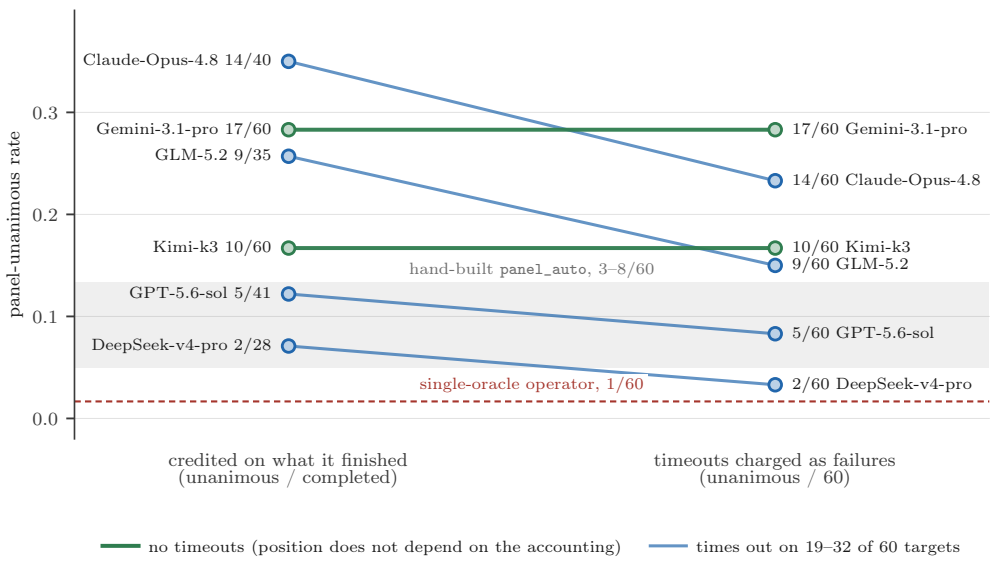

Figure 15: **Why Table 18 is alphabetical.** The same six agents under the two defensible accountings of a timeout. Credit each model only on the targets it finished and Claude-Opus-4.8 leads; charge timeouts as failures and Gemini-3.1-pro does. The crossing lines are the finding: an ordering that reverses under a bookkeeping choice is not an ordering, so we report none. The two flat lines are the models that never time out, and their position is the only one here that does not depend on the choice. What survives both accountings is the single claim this appendix makes: all six clear the single-oracle operator and the decidable floor (`model_sensitivity.py`).

`inverse_fold_cf`, `fold_pkiss_pairs` and `fold_probknot_pairs` (crossing-aware pair sets), `target_pairs`, `rand_seq` and `enforce_pairs`. Each round the agent's code is executed on a 16-target training pool disjoint from every evaluation pool, the per-target outcomes are returned as feedback, and the agent rewrites. The best round by training score is frozen and adjudicated once on the development/evaluation 60; the frozen source is archived at `runs/cc_ascent/invent_operator/invented_design.py`.

Provenance and its limit. Runs call frontier models through a single provider (OpenRouter) at temperature 0.4. Each `invent_log.json` records the provider-qualified model identifier, per-round prompt and completion tokens, per-round training score, and every compile or runtime error verbatim; each frozen `design` program is stored in full in `result.json`. What the logs cannot contain is a provider-side version hash, because the endpoint exposes none: a frontier model behind a routed alias is not a fixed artifact. That is a reproducibility limit of the invention step and of nothing measured in this paper, since every number here re-executes frozen source rather than re-querying a model.

The agent's training solve rate rises as it folds oracle feedback into successive rewrites, and the frozen best round measures $9/60$ under all three predictors. The analogous frozen results for all six frontier models are in Table 18, and every one peaks by round ≤ 5.

Design limits of the six-model comparison. One temperature-0.4 draw per model, no seed replication, and the two models later replicated span 10–18 on this metric across seeds, a range wider than most gaps in Table 18. Four of the six also time out on 19–32 of 60 targets; that table charges the timeouts as failures and shows the ordering flipping when they are not (`model_sensitivity.py`). The experiment therefore supports one statement only, that six single-run agents produced nonzero cross-predictor consistency under a common invention protocol, and supports no ordering among them. A model comparison would need several draws per model at several temperatures, a shared compute budget, and a clustering of operator sources.

Table 18: **An agent invents the operator, across six frontier models** (three open, three closed). Each agent, given only primitives and the two crossing oracles, writes and revises a `design` operator; the frozen best is adjudicated by the panel on the development/evaluation 60. Reference points: hand-built `panel_auto` 34–37 / 3–8; single-oracle 43 / 1; CF floor 0. **Rows are alphabetical, not ranked**, and timeouts are charged as failures against the full 60. Crediting models only on what they finished would rank Claude-Opus-4.8 first (14/40); charging timeouts ranks Gemini-3.1-pro first (17/60). That the order flips under an accounting change is why we claim no order, and Fig. 15 is that flip. What survives both is the only claim made here: all six clear the single-oracle operator and the floor (`model_sensitivity.py`).

agent	weights	train /16	held-out pKiss∧ProbKnot	**panel-unanimous** /60
Claude-Opus-4.8	closed	12	40/60	14
DeepSeek-v4-pro	open	5	26/60	2
Gemini-3.1-pro	closed	12	38/60	17
GLM-5.2	open	12	35/60	9
GPT-5.6-sol	closed	12	41/60	5
Kimi-k3	open	12	42/60	10

K Oracle Configurations, Hyperparameters, Compute, and Verifier Stringency

Oracle configurations, all deterministic and replayable with `DATAPATH` pinned to the RNAstructure data tables: Ω_{cf} is ViennaRNA Turner-2004 MFE at 37°C (env `rna`, no timeout); pKiss is `pkiss mfe` (Janssen & Giegerich, 2015) (env `pk`, 60 s); ProbKnot is RNAstructure MEA over the partition ensemble (Bellaousov & Mathews, 2010) (env `rnastr`, 60 s); ShapeKnots is RNAstructure MFE-based run *de novo* with no SHAPE data (Hajdin et al., 2013) (env `rnastr`, 120 s).

Operator hyperparameters. REMC floor: wall 30 s. Design operators (fixed across all targets, with size-independent budgets): skeleton budget 12–20 s, repair budget 90–120 s, $\leq$ 800–1500 repair iterations, ProbKnot gate at $\text{pk_bp} \leq 3$. **Compute.** A single 256-core host; target-level parallelism at $\text{NWORKERS}=60$ (the design operators are single-process, so this is safe); the entire sweep suite is search-only and incurs *no* model-inference cost, except the invention agent (App. J), a few dollars of GLM-5.2.

Positive-side stringency (the verifier ladder), and one rung beyond panel-unanimity. Panel-unanimity is already stronger than "the target pairs appear somewhere": it requires the target to *equal* each method's predicted structure, the MFE under pKiss and ShapeKnots and the maximum-expected-accuracy structure over a partition-function ensemble under ProbKnot. Those are related but not identical notions of a single dominant fold, since ProbKnot's MEA is not a unique-MFE criterion, so panel-unanimity sits near the "unique/dominant fold" region of the exact $\subseteq$ unique-MFE $\subseteq$ dominance ladder and not exactly on one rung, agreed by three predictors that are thermodynamically related rather than statistically independent.

We climb one rung further and measure it (`runs/cc_ascent/positive_stringency.py`). Enumerating pKiss suboptimals for the mixed-panel-consistent designs and taking the free-energy gap Δ from the target to the best alternative fold: the target is the pKiss MFE for all 17 designs whose enumeration completed within budget, dominant by median Δ=1.2 kcal/mol (mean 1.44), with 14/17 clearing a ≥ 1 kcal margin (Fig. 16). Eight longer designs timed out, and since timeout correlates with length this rung covers the shorter adjudicated designs only. The gap is a property of the pKiss energy model, not a model-independent thermodynamic fact. Within that scope the positive verdict is not a marginal coincidence of pair sets: the target is the free-energy-minimum fold by a real if modest margin.

Two further strengthenings are computable with the installed tooling: a Boltzmann-probability threshold (RNAstructure's `partition` and `ProbabilityPlot`), and a fourth, non-thermodynamic predictor. Physical confirmation by in-vitro probing is the only route past model-relativity and is out of scope here.

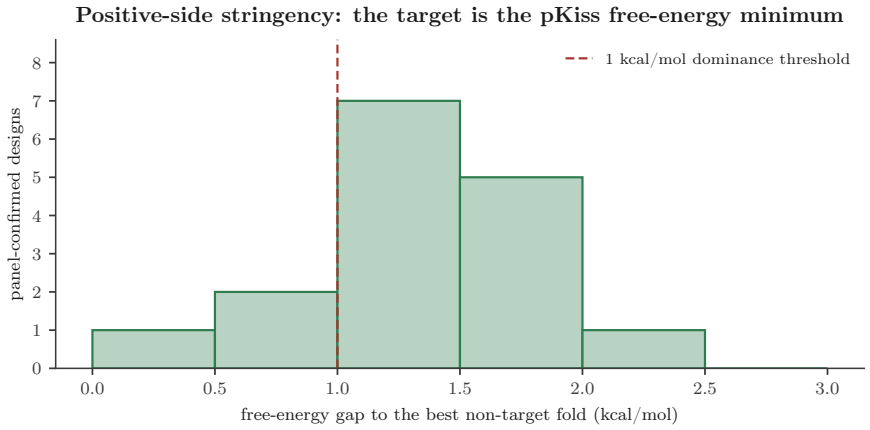

Figure 16: **One rung above panel-unanimity, as a distribution rather than a median.** Free-energy gap from the target fold to the best non-target pKiss suboptimal, for the 17 panel-confirmed designs whose enumeration completed. A median of 1.2 kcal/mol is compatible with a bimodal picture in which some designs are marginal; the distribution shows it is not, with $14/17$ clearing 1 kcal and the left tail thin rather than heavy. The scope limits stated in the text still apply: eight longer designs timed out, timeout correlates with length, and the gap is a property of the pKiss energy model, not a model-independent thermodynamic fact (`positive_stringency.py`).