BOUNDING WHAT LEAN GETS TO SEE: CANDIDATE EXPOSURE IN PROOF-ACTION RANKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Tactic rankers determine which proof actions Lean gets to check. Trace matching measures recovery of one recorded action; candidate exposure measures whether the submitted list contains any action Lean accepts. We derive exposure guarantees for similarity retrieval with tactic-family priors. A soft prior of weight λ preserves within-family order and can reverse only comparisons within a similarity gap of λ. Counting these rivals gives a sufficient top-k retention condition under any family prediction; accounting for their shared probabilities gives the exact worst-case rank. Hard routing instead places entire families ahead of a candidate, regardless of similarity. Experiments use 3,723 proof steps from 1,702 theorems in a curated mathlib4 subset. Hard routing moves the first trace-family candidate from mean rank 41 to 531, while soft and unguided ranking have nearly identical trace exact@5 (0.170 and 0.169). Reanalysis of fixed top-five lists with corrected, target-excluding Lean outcomes on 497 matched states gives Accept@5 of 0.412 for unguided retrieval, 0.408 for the soft prior, and 0.342 for hard routing. The hard-minus-unguided difference is -7.04 percentage points (95% theorem-cluster interval $[-11.80, -2.59]$); the soft difference is -0.40 points ($[-1.70, 0.84]$). Hard routing leads at the first candidate but falls behind by the third. For unguided retrieval, 139 of 205 accepted lists (67.8%) contain no trace hit. The checking cutoff is therefore part of the ranking objective: family predictions should preserve accepted alternatives at the list length the prover actually uses.

1 INTRODUCTION

Ranking determines which proof actions reach Lean. A tactic outside the submitted list receives no formal feedback, even when it would execute successfully (de Moura & Ullrich, 2021). With a fixed number of checks, the first question is whether the list contains any acceptable action. This event, termed *candidate exposure* by Kamat & Wang (2026), connects a ranking decision to the proof assistant's next available moves.

Human proofs record one continuation at each state. Lean may also accept other rewrites, simplifications, constructions, or closing terms. Exact trace matching counts recovery of the recorded choice and misses these alternatives. Execution tests the retrieved choices directly, including whether they leave residual goals. A ranker can consequently improve trace recovery, action applicability, and immediate closure by different amounts, or in opposite directions.

A prediction of `rw`, `simp`, `exact`, or another tactic family changes which retrieved actions receive those checks. Hard routing places whole families in a predicted order. A soft prior adds a bounded family-dependent score to similarity. The rules preserve within-family order but can produce very different cross-family ranks. We ask when an accepted action survives this change at the top-k boundary, even if the family prediction is wrong.

The answer depends on the distribution of candidates around the list boundary. A bounded family bonus restricts reversals to a local similarity interval. Counting the rivals in this interval bounds displacement and gives a sufficient condition for retaining an accepted action. Accounting for the probability shared among rival families gives the exact worst-case rank. Hard routing follows a different rule: every record in a preceding family contributes to rank displacement, however low its similarity.

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.

We examine these effects on 3,723 mathlib4 proof steps. Across six theorem-disjoint splits, hard routing moves the first trace-family record from mean rank 41 to 531; soft ranking leaves trace exact@5 almost unchanged. Corrected, target-excluding Lean outcomes then evaluate the submitted lists themselves. On 497 shared states, unguided retrieval exposes an accepted top-five action in 205 states, soft ranking in 203, and hard routing in 170. Hard routing leads these three methods at the first check but falls behind by the third. On the panel shared by five rankers, family interleaving raises acceptance while reducing immediate closure.

Contributions. We derive candidate-retention conditions and an exact worst-case rank over family distributions, characterize displacement under hard routing, and relate the resulting lists to Lean acceptance and closure. These results extend the workshop study of Kamat & Wang (2026), which introduced candidate exposure, the retrieval collection, the additive score, and the pairwise gap observation. The present paired analysis uses corrected target-excluding outcomes and supersedes its acceptance estimates. Section 4 describes the label correction; the supplement gives the proofs and extended comparisons.

2 Candidate exposure and family-aware ranking

Candidate records. At a proof state s, let C_s be a finite collection of retrieved training records. Each record $a \in C_s$ carries a tactic string $\tau(a)$, a stored proof state, and a family $f(a)$. Distinct records can carry the same string. We retain these duplicates because the studied rankers rank records; five positions can therefore contain fewer than five distinct actions. Let $q_s(a) \in [0, 1]$ be the cosine similarity between query and stored state under nonnegative term-frequency–inverse-document-frequency (TF–IDF) representations. A fixed record index resolves residual ties.

Trace recovery, acceptance, and closure. Write $a_s^\star$ for the recorded next tactic. For a checker environment E, let $A_E(s)$ be the set of candidate records whose tactics execute without a tactic error at the reconstructed state. The environment includes declarations, attributes, local context, and the checking protocol. For a ranking π and a positive integer cutoff k, let $\mathrm{Top}_k(\pi, s)$ be its first $\min(k, |C_s|)$ records. Define

$$X_k(\pi, s) = \mathbf{1}\{\exists a \in \mathrm{Top}_k(\pi, s) : \tau(a) = a_s^\star\}, \tag{1}$$

$$V_k^E(\pi, s) = \mathbf{1}\{\mathrm{Top}_k(\pi, s) \cap A_E(s) \neq \varnothing\}. \tag{2}$$

Equality in Equation (1) uses the normalization specified for each experiment. Exact@k and Accept@k average these indicators across states. Let $f_s^\star$ denote the family of the recorded tactic; Family@k tests whether some exposed record has $f(a) = f_s^\star$. Other families may also contain acceptable actions. Acceptance allows residual goals and actions that make no progress. Close@k instead requires an exposed action to leave no residual goals in the reconstructed state; eventual completion of the original theorem remains a separate endpoint.

Soft scores and the hard endpoint. Let $p_s(f) \in [0, 1]$ be a predicted distribution over tactic families, with zero probability assigned to unseen families. For a weight $\lambda \geq 0$, rank by

$$Q_\lambda(s, a) = q_s(a) + \lambda p_s(f(a)), \tag{3}$$

then by raw similarity, then by fixed record index. Denote the resulting one-based rank by $r_\lambda(a; s)$. At $\lambda = 0$ this is unguided similarity retrieval. Hard routing ranks first by a total predicted family order ρ_s, then by similarity and record index. It prioritizes every candidate in an earlier family over every candidate in a later family.

For distinct represented-family probabilities, the soft order eventually equals hard routing as $\lambda \to \infty$. Tied probabilities retain cross-family similarity order in the numerical limit; the implemented hard router resolves the family tie first. The total order ρ_s specifies that choice. Both rules preserve within-family order, while finite soft weights also bound cross-family score changes.

3 Rank displacement and candidate retention

Fix a state, candidate collection, and tie rule. The bounds hold for every predicted family distribution p_s. Full proofs and counterexamples appear in the Supplementary Material.

Theorem 1 (Within-family invariance)**.** *For any two candidates a, b with $f(a) = f(b)$, their relative order is identical under unguided retrieval, every finite soft weight, and hard routing.*

Proof. The family terms in $Q_\lambda(s, a) - Q_\lambda(s, b)$ cancel. Hard routing assigns both records the same family priority. In either case, similarity and the common residual tie rule determine their order. □

Family prediction therefore acts entirely through cross-family comparisons. Promoting a family can expose one of its records, but the records ahead of it in that family stay ahead. Its argument list and within-family position are fixed even under a perfect family prediction.

For a candidate a, define the set of initially lower-ranked rivals within the perturbation range,

$$B^+_\lambda(a; s) = \{b \in C_s : r_0(b; s) > r_0(a; s),\ 0 \le q_s(a) - q_s(b) \le \lambda\}. \tag{4}$$

Closed interval endpoints include all tie cases. Removing same-family candidates tightens the bound.

Theorem 2 (Local displacement bound)**.** *If $q_s(a) - q_s(b) > \lambda$, then a precedes b under the soft ranking for every predicted family distribution. Consequently,*

$$r_\lambda(a; s) \le r_0(a; s) + |B^+_\lambda(a; s)|. \tag{5}$$

An analogous lower bound subtracts the number of initially preceding candidates whose similarity is at least $q_s(a)$ and differs from it by at most λ.

Proof. The family-probability difference lies in $[-1, 1]$, so $Q_\lambda(s, a) - Q_\lambda(s, b) \ge q_s(a) - q_s(b) - \lambda$. Every record newly preceding a must therefore belong to $B^+_\lambda(a; s)$. Adding at most their number to the original rank proves Equation (5); records that move behind a can only reduce this rank. □

Corollary 3 (Sufficient exposure condition)**.** *Fix an environment E. If some accepted candidate $a \in C_s \cap A_E(s)$ satisfies*

$$r_0(a; s) + |B^+_\lambda(a; s)| \le k, \tag{6}$$

then $V^E_k(\pi_\lambda, s) = 1$ for every predicted family distribution.

Applying the rank bound to the accepted record proves the corollary. The condition combines its baseline rank, the number of crossable rivals, and its checker outcome. At rank k, a single promoted rival can remove the record; farther inside the list, several rivals must pass it. Acceptance of the retained record is established by Lean.

Displacement grows with the number of nearby rivals. For any positive weight, arbitrarily many records of one other family can lie in an interval narrower than λ below a candidate. Assigning probability one to that family lets all these records pass it. A wide similarity gap protects the record under the same prediction error. The score scale and the local density of candidates jointly determine the rank effect.

Margin at the cutoff. The count condition has an equivalent boundary-margin form before same-family rivals are removed. Assume $k < |C_s|$, and let $q_{s,(k+1)}$ be the similarity of the record at baseline rank $k + 1$. If an accepted baseline top-k record a satisfies

$$q_s(a) - q_{s,(k+1)} > \lambda, \tag{7}$$

then no baseline record outside the top k can pass it. The other baseline top-k records contribute at most $k - 1$ predecessors, so a stays exposed. For the unrefined neighborhood in Equation (4), this is the same sufficient condition written as a boundary gap: if the first outside record lies within λ, every intervening lower-ranked record also belongs to that neighborhood, forcing the rank bound above k. Removing same-family rivals can sharpen the count bound, because those rivals cannot cross a. For $k = |C_s|$, exposure is invariant under any permutation of the pool.

For a fixed prediction, let $\Delta_s = \max_f p_s(f) - \min_f p_s(f)$ over represented families. The radius $\lambda\Delta_s$ replaces λ in the local bounds because the proof uses only the range of the family bonus. A uniform prior has $\Delta_s = 0$ and leaves every rank unchanged. A point-mass prior on one of at least two represented families has $\Delta_s = 1$. The probability spread sets the score perturbation, and the density of nearby records converts it into rank movement.

Exact worst-case rank. The neighborhood bound treats each rival as independently promotable. A probability distribution imposes a further restriction: promoting several different families requires sharing a total probability of one. Let $\mathcal{G}_a$ be the represented families other than $f(a)$, and, for a similarity threshold $t \geq 0$, define

$$N_g(t; a) = \big|\{b \in C_s : r_0(b; s) > r_0(a; s),\ f(b) = g,\ q_s(a) - q_s(b) < t\}\big|.$$

The strict inequality matters because an initially lower-ranked record loses an equal-score comparison under our tie rule. Let x_g allocate probability to rival family g, and write $r_{\lambda,p}$ for the rank under distribution p.

Theorem 4 (Exact worst-case rank)**.** *For every finite weight $\lambda \geq 0$, the maximum over all represented-family distributions is*

$$\max_p r_{\lambda,p}(a; s) = r_0(a; s) + \max_{\substack{x_g \geq 0\ (g \in \mathcal{G}_a) \\ \sum_g x_g \leq 1}} \sum_{g \in \mathcal{G}_a} N_g(\lambda x_g; a). \tag{8}$$

For an empty $\mathcal{G}_a$, the added term is zero.

To maximize the rank, probability can be moved away from $f(a)$ to rival families without improving any comparison for a. Every baseline predecessor then remains ahead, and N_g counts exactly the additional predecessors in family g. The supplement gives the complete proof, including attainment and ties.

For example, take a three-record pool: a has similarity 0.80 and its two rivals, each in another family, have similarity 0.65. With $\lambda = 0.25$, each rival can pass a separately, but both cannot: each would require more than 0.60 probability when $f(a)$ receives none. The worst rank is two, whereas the separate-rival count allows three. Thus an accepted a remains exposed at cutoff two under every prediction. More generally, bounding the right side of Equation (8) by k is necessary and sufficient for retaining this particular record. It remains only sufficient for exposure, which may be preserved by different accepted records under different predictions.

A finite form of the exact rank. The maximization can be written using only the sorted similarity gaps. For each $g \in \mathcal{G}_a$, let ℓ_g count its records below a in the baseline order, and let $d_g(m)$ be their mth smallest gap $q_s(a) - q_s(b)$, with $d_g(0) = 0$. For $\lambda > 0$, Equation (8) becomes

$$\max_p r_{\lambda,p}(a; s) = r_0(a; s) + \max\left\{\sum_g m_g :\ m_g \in \{0, \ldots, \ell_g\},\quad \sum_g d_g(m_g) < \lambda\right\}. \tag{9}$$

Promoting $m_g > 0$ records of family g requires probability strictly greater than $d_g(m_g)/\lambda$ when $f(a)$ receives zero. The chosen promotions are jointly feasible exactly when these thresholds sum to less than one. The strict inequality implements the raw-similarity and record-index tie breakers. At $\lambda = 0$, all ranks equal their baseline values.

This form compares allocations across families directly. Many closely spaced rivals in one family can cross together under one bonus; equally close rivals spread over several families compete for probability. The local count bound captures the score gaps, while Equation (9) also captures how those gaps are grouped.

Proposition 5 (Hard-routing displacement)**.** *Let $j_f(a; s)$ be the similarity rank of a among records in family $f = f(a)$, and let $n_g(s)$ be the number of records in family g. Under hard routing,*

$$r_{\mathrm{hard}}(a; s) = j_f(a; s) + \sum_{g:\rho_s(g) < \rho_s(f)} n_g(s). \tag{10}$$

In particular, for arbitrarily large candidate pools, an accepted candidate at similarity rank one can be displaced beyond any fixed cutoff by an incorrect family prediction, even when its similarity is one and every preceding family's similarity is zero.

Put the accepted record in one family and at least k rejected records in a family placed first by ρ_s. Those records occupy the whole submitted list, including when their similarities are zero and the accepted record's similarity is one. Hard routing thus converts family population directly into displacement. Exposure survives a wrong trace-family prediction precisely when an accepted alternative still reaches the selected positions.

Proposition 6 (Fixed-list reranking ceiling). *For a fixed exposed set $L \subseteq C_s$, no reranking of L can expose an accepted candidate when $L \cap A_E(s) = \varnothing$. If every record in L is checked, exposure is invariant under reranking.*

A second-stage predictor can bring an accepted record to the front of its input list. The best exposure achievable by reranking is fixed by the membership of that list. First-stage selection therefore determines which accepted actions any subsequent reranker can recover.

4 Experimental design

Corpus and splits. The study uses 3,723 checked proof steps from 1,702 theorems in 46 files of a curated mathlib4 subset (The mathlib Community, 2020; Yang et al., 2023), with 81 tactic families. The collection covers arithmetic, mixed-library, and proof-dense modules. All steps of a theorem stay in one split. The trace-ranking study uses six repeated theorem splits with 60% training, 10% validation, and 30% test theorems; the split identifiers are given in the supplement. Different theorems from the same source file can occur in different splits. The six-split exact-match metric compares stored tactic strings directly; trace events on the repaired execution panel instead collapse whitespace before comparison.

Rankers. All principal rankers use the same state-only TF–IDF cosine similarities and training-record candidates within each experiment. The family predictor is logistic regression on TF–IDF features with balanced class weights. Soft weights are selected by validation trace exact@5, breaking ties by exact@1, family@5, and then the smaller weight. Hard routing uses the full descending family order. We also report reciprocal-rank fusion (RRF) of family and similarity ranks, round-robin interleaving of the top three families, and trace-family oracle routing. The oracle is given the recorded family, which need not contain every acceptable action.

Frozen execution lists. The one-step study uses a separate 70/30 theorem split and 500 sampled test states, with five candidate records per ranker. Reconstructing its soft ranking at $\lambda = 0.25$ recovers all 2,500 candidate positions. The execution study therefore evaluates this fixed weight, while the trace study evaluates validation-selected weights. We keep every submitted list unchanged and replace only its checker outcomes. All paired comparisons use lists from the same execution split.

Corrected checker labels. We match the frozen lists to later repaired executions by state identifier and whitespace-normalized tactic. The primary checker reconstructs the source prefix immediately before the target declaration, preserving earlier declarations while excluding the target and its attributes. Candidate-specific state markers and corrected physical-line attribution separate successful execution from tactic errors. Acceptance allows residual goals. Closure is measured immediately after the candidate executes, before completion of the surrounding test file. Acceptance is thus one-step applicability in the reconstructed, target-excluding environment.

The corrected outcomes replace two defects in the original labels: completed-module imports exposed the target theorem, and a source-line mapping error assigned some multiline failures to the wrong candidate. Both repairs enter the present reanalysis. We retain the 497 states with five matched outcomes per principal ranker and successful environment construction in every stored replay condition. One state lacks hard-routing outcomes and two have reconstruction errors. This common set also supports the supplementary environment comparison. We bound the effect of all three exclusions using the original 500-state denominator.

Uncertainty and endpoints. We report paired exposure differences on the common states. The primary intervals resample theorem clusters, retaining all selected states of each sampled theorem to account for within-proof dependence. The supplement also reports exact state-level McNemar tests. The analysis is retrospective: lists and soft weights were fixed before the corrected outcomes were compared. Accept@k measures one-step applicability and Close@k measures immediate closure of the reconstructed state.

Table 1: Hard routing reduces trace recovery across six theorem splits; soft and unguided scores remain close. Entries are split means, with exact matching of stored strings. Family rank is conditional on that family being present. The oracle receives the trace family, not accepted-action labels.

Ranker	Family@5	Exact@1	Exact@5	Family rank
Unguided	0.600	0.120	0.169	40.8
Soft family	0.589	0.123	0.170	42.4
Hard family	0.130	0.069	0.082	531.3
Family RRF	0.546	0.111	0.169	94.6
Top-three interleaving	0.231	0.065	0.111	102.9
Trace-family oracle	0.988	0.199	0.245	1.0

Table 2: Hard routing loses top-five acceptance despite its higher Accept@1. Corrected target-excluding outcomes use 497 shared states (278 theorems). Differences from unguided and 95% theorem-cluster bootstrap intervals are in percentage points.

Ranker	Accept@1	Hits@5	Accept@5	Δ (pp)	95% CI
Unguided	0.223	205/497	0.412	Reference	–
Soft family	0.239	203/497	0.408	-0.40	[-1.70, +0.84]
Hard family	0.288	170/497	0.342	-7.04	[-11.80, -2.59]

5 Results

5.1 Hard routing displaces the trace family

Hard routing reduces family@5 from 0.600 to 0.130 and exact@5 from 0.169 to 0.082 across the six theorem splits (Table 1). The mean rank of the first trace-family record rises from 40.8 to 531.3, conditional on that family being present in the training pool. Equation (10) gives the mechanism: every record in a higher-priority family precedes the queried family. Prediction errors that promote populous families incur a correspondingly large rank penalty.

Soft ranking stays close to unguided retrieval: its mean first trace-family rank is 42.4 and its exact@5 is 0.170. Reciprocal-rank fusion reaches exact@5 of 0.169 but places the first trace-family record at mean rank 94.6. These aggregate trace scores describe recovery of the recorded family and string. The next comparison checks the submitted actions in Lean.

The trace-family oracle raises exact@5 to 0.245, leaving a mean miss rate of 75.5% across splits. Correct family information leaves both candidate availability and within-family retrieval unresolved. A recorded string may be absent from the training pool or remain below records with different arguments or premises. Theorem 1 fixes the latter ordering even under the oracle prediction.

5.2 Lean acceptance reverses the ranking as the list grows

Unguided retrieval exposes an accepted top-five action on 205 of the 497 matched states, soft ranking on 203, and hard routing on 170 (Table 2). Hard routing loses 35 exposed states in aggregate, a difference of -7.04 percentage points with a 95% theorem-cluster interval of $[-11.80, -2.59]$. The soft difference is -0.40 points, with interval $[-1.70, 0.84]$.

The paired counts show how each method changes the baseline's successful lists. Soft ranking retains 199 of the 205 unguided successes, loses six, and gains four elsewhere. Hard routing retains 135, loses 70, and gains 35. It changes the binary top-five outcome on 105 states, compared with ten for soft ranking. Thus the hard-routing deficit combines substantial losses with gains on a smaller set of states.

Cutoff and closure. Hard routing leads at the first candidate, with 143 accepted states versus 111 unguided and 119 soft (Figure 1, left). By the third candidate, the ordering reverses: hard routing reaches 159 states, unguided retrieval 173, and soft ranking 175. Extending the list from one to five adds 27 successful states for hard routing, 94 for unguided retrieval, and 84 for soft ranking.

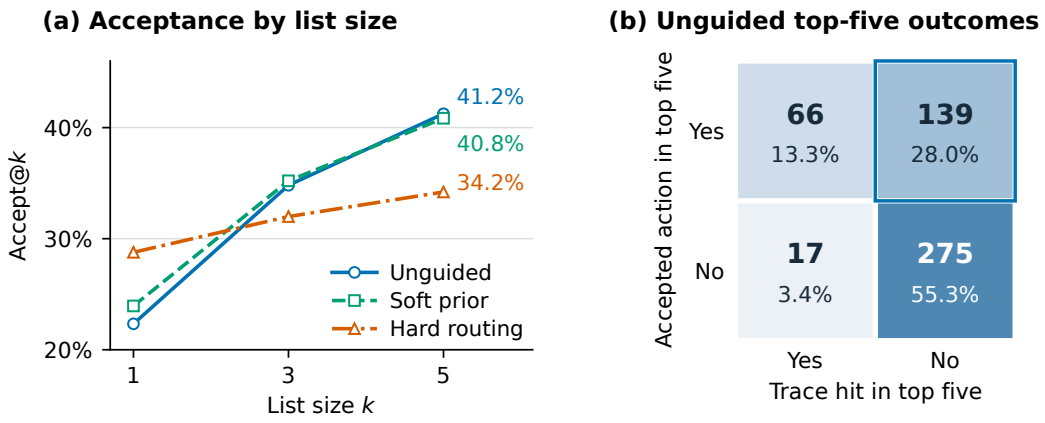

Figure 1: Checking more candidates reverses the method ordering. Left: hard routing has the highest acceptance at one candidate and the lowest at three and five among the principal rankers. Right: unguided top-five lists on the same 497 states; 139 states have an accepted action without a trace hit, while 17 have a trace hit without acceptance. Each state contributes once per panel.

Table 3: Cutoff and endpoint sensitivity on the same 497 states. Accept includes non-closing and unchanged states; Close requires no residual goals after a retrieved action. Each entry is the fraction of states with at least one qualifying candidate in the list.

Ranker	Accept@1	Accept@3	Accept@5	Close@5
Unguided	0.223	0.348	0.412	0.155
Soft family	0.239	0.352	0.408	0.157
Hard family	0.288	0.320	0.342	0.107

These increments count newly exposed states; several accepted actions in one list still contribute one success.

Immediate closure gives a second distinction (Table 3). At cutoff five, unguided, soft, and hard ranking expose a closing action in 77, 78, and 53 states, respectively. An accepted action may transform the goal, leave it unchanged, or close it. The first-candidate acceptance advantage of hard routing therefore coexists with lower top-five closure.

List overlap and exclusions. Soft ranking returns the same five ordered, whitespace-normalized strings as unguided retrieval on 276 states and the same distinct-string set on 383 states. It changes the action set on 114 states, of which ten change exposure. Hard routing preserves the distinct-string set on 18 states. These comparisons measure changes to the submitted actions separately from changes to their acceptance outcomes.

The hard-routing deficit persists under every assignment of outcomes to the three omitted states. Each missing paired difference lies in $[-1, 1]$, so the full 500-state difference lies in $[(-35-3)/500, (-35+3)/500]$, or $[-7.6, -6.4]$ percentage points. The corresponding soft interval is $[-1.0, 0.2]$ points. These deterministic exclusion bounds complement the sampling intervals in Table 2.

5.3 Most accepted lists miss the trace

Unguided retrieval exposes an accepted action without a trace hit on 139 of 497 states (27.97%). These states account for 67.8% of its 205 accepted lists. Soft ranking has 138 acceptance-without-trace states (27.77% of the panel) among its 203 accepted lists. Both methods recover the recorded string on 83 states (16.70%). Figure 1 (right) gives the complete joint distribution for unguided retrieval. Sixty-six states have both a trace hit and an accepted list. Adding the 139 acceptance-only states yields 205 accepted lists; adding the 17 trace-only states yields 83 trace hits. The remaining 275 states have neither event. The discrepancy therefore runs in both directions.

Table 4: Interleaving improves acceptance while reducing closure on the 491-state panel shared by five rankers. Differences in Accept@5 from unguided are in percentage points; theorem-cluster intervals have no simultaneous adjustment.

Ranker	Accept@5	Close@5	Δ (pp)	95% CI
Unguided	0.411	0.155	–	–
Soft family	0.409	0.157	-0.20	[-1.37, +0.99]
Hard family	0.342	0.106	-6.92	[-11.68, -2.25]
Family RRF	0.420	0.153	+0.81	[-1.67, +3.35]
Top-three interleaving	0.456	0.124	+4.48	[+0.65, +8.37]

Table 5: Hard routing concentrates five-record lists in fewer families and distinct actions. Tactics are whitespace-normalized; the last two columns count states with one family or five distinct tactics, respectively, among the same 497 states.

Ranker	Mean tactics	Mean families	One family	Five tactics
Unguided	4.43	3.10	36	343
Soft family	4.32	3.05	51	328
Hard family	3.23	1.24	388	184

An accepted non-trace tactic may also occur beside a trace match. This happens on 25 states: together with the 139 acceptance-only states, they account for all 164 lists containing an accepted alternative. The distinction matters when interpreting an "alternative": its presence and the absence of the recorded tactic are different properties of the same list.

Seventeen lists contain the recorded string but have no accepted action under reconstructed-state checking. Pretty-printed goals can omit parts of the original elaborator state, and transferred tactics may refer to unavailable local names. Executing the recorded tactic tests whether reconstruction preserves that continuation; recovering its string alone does not. Both events must be measured on the same states and checker environment.

An accepted simplification excluded by hard routing. Consider a selected state with natural-number variables n, a, b, c and goal

$$bn + a \equiv c \pmod{n} \quad \Longleftrightarrow \quad a \equiv c \pmod{n}.$$

The recorded tactic is `rw [add_comm, add_mul_modulus_modEq_iff]`. Unguided retrieval places `simp [ModEq]` first; the corrected target-excluding replay leaves no residual goals. None of its five retrieved records matches the trace. Hard routing selects five different training records that all carry `rfl`; Lean rejects all five in the same environment.

5.4 Family coverage and repeated actions

Top-three interleaving exposes an accepted action on 224 of the 491 states shared by all five rankers, compared with 202 for unguided retrieval (Table 4). Its gain is 4.48 percentage points, with an individual theorem-cluster interval of $[0.65, 8.37]$. RRF changes acceptance by 0.81 points ($[-1.67, 3.35]$). Each comparison uses the same 491 states; the additional rankers account for the smaller panel relative to the primary analysis. The four secondary intervals are unadjusted for simultaneous inference.

Interleaving preserves each family's similarity order and cycles through the three highest-priority families. Whenever several selected families still contain records, each contributes one per round. The accepted-action gain accompanies lower immediate closure: interleaving has 61 closing lists versus 76 for unguided retrieval on this panel. At the same cutoff, applicability and closure favor different lists.

Distinct actions per list. Five retrieved records often contain fewer than five tactics. On the primary panel, unguided retrieval averages 4.43 distinct whitespace-normalized strings across 3.10 families; soft ranking averages 4.32 strings across 3.05 families (Table 5). Hard routing averages 3.23 strings across 1.24 families. Its five positions come from a single family on 388 of 497 states, compared

with 36 unguided and 51 soft lists. Five distinct tactics appear in 184 hard lists, 343 unguided lists, and 328 soft lists.

Repeated strings consume positions without adding an action to the accepted set. Family concentration and repetition occur together in the hard lists, so their individual contributions require a separate comparison. Deduplication would replace repeated records with additional actions and change which candidates Lean checks. Here the record-based evaluation exposes both restrictions directly: fewer families reach the list, and some positions repeat the same tactic.

6 Related work

Learned theorem provers use checker feedback to select tactics and guide search. TacticToe combines learned tactic selection with search; CoqGym and HOList provide interactive learning environments; HyperTree Proof Search learns from search experience (Gauthier et al., 2021; Yang & Deng, 2019; Bansal et al., 2019; Lample et al., 2022). LeanDojo retrieves accessible premises for language-model proving, while DeepSeek-Prover-V1.5 uses formal feedback for reinforcement learning and tree search (Yang et al., 2023; Xin et al., 2025). Local look-ahead evaluates intermediate actions through verifier feedback (Rajaee et al., 2025). These systems make the selection of actions for checking a recurring decision. We study how family scores choose that list from a fixed candidate pool.

Learning to rank treats the proof state as a query and candidate tactics as competing items (Rudin & Wang, 2018). A family bonus assigns the same value to many items, leaving their internal order fixed. This shared score makes the movement between groups explicit: local gaps control soft-score reversals, family populations determine hard-routing ranks, and the probability sum couples promotions across families. The invariance and local bounds apply to bounded group-constant bonuses more generally; the exact characterization uses a normalized family distribution.

The candidate-exposure study of Kamat & Wang (2026) provides the direct empirical starting point. Its gap observation identifies which pairwise comparisons can change. The rank conditions developed here determine whether enough of those changes can occur together to remove an accepted record from the submitted list. The corrected execution analysis then measures acceptance of the selected actions in an environment that excludes the target theorem.

7 Discussion and conclusion

Family information changes proof-action ranking through both score gaps and the number of records promoted. A bounded score restricts the eligible comparisons; shared probabilities restrict which promotions can occur together. Hard routing places entire families first and can move a high-similarity accepted action arbitrarily far down a growing pool. At a fixed weight, each cross-family score difference changes by λ times the difference in predicted family probabilities. A concentrated prediction can therefore reverse comparisons that a diffuse prediction preserves, even when both predict the same leading family; choosing the weight requires the probability spread as well as the local similarity gaps. These formulas identify the distinct effects of family prediction, candidate density, and family population.

The experiments evaluate these choices on a curated mathlib4 collection with lexical models, theorem-disjoint splits that share source files, and reconstructed one-step states. Acceptance includes non-closing and unchanged states. Candidate arguments are transferred as stored, and repeated strings occupy separate record positions. Full-pool scores are needed to measure how often the retention conditions hold; the fixed-list reanalysis measures acceptance under the ranking rules. File-disjoint evaluation, candidate deduplication, and successor-state search would test the corresponding changes to generalization, action coverage, and proof completion.

At five candidates, hard routing lowers acceptance by 7.04 percentage points relative to unguided retrieval, while the soft difference is -0.40 points. At one candidate, hard routing leads. Interleaving raises top-five acceptance on the five-method panel but lowers immediate closure. The choice of ranker therefore depends on the actions and endpoint that the checker is asked to evaluate. Candidate exposure measures this choice directly. Trace recovery, one-step acceptance, and proof completion should be reported as separate outcomes.

REFERENCES

Kshitij Bansal, Sarah Loos, Markus Rabe, Christian Szegedy, and Stewart Wilcox. HOList: An environment for machine learning of higher order logic theorem proving. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 454–463. PMLR, 2019. URL https://proceedings.mlr.press/v97/bansal19a.html.

Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. In *Automated Deduction – CADE 28*, volume 12699 of *Lecture Notes in Computer Science*, pp. 625–635. Springer, 2021. doi: 10.1007/978-3-030-79876-5_37. URL https://link.springer.com/chapter/10.1007/978-3-030-79876-5_37.

Thibault Gauthier, Cezary Kaliszyk, Josef Urban, Ramana Kumar, and Michael Norrish. TacticToe: Learning to prove with tactics. *Journal of Automated Reasoning*, 65(2):257–286, 2021. doi: 10.1007/s10817-020-09580-x. URL https://link.springer.com/article/10.1007/s10817-020-09580-x.

Shivangi Kamat and Yangshuai Wang. Before Lean checks: Candidate exposure in proof-action ranking. In *3rd AI for Math Workshop at the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=HUIJyoVgT3.

Guillaume Lample, Timothée Lacroix, Marie-Anne Lachaux, Aurélien Rodriguez, Amaury Hayat, Thibaut Lavril, Gabriel Ebner, and Xavier Martinet. HyperTree proof search for neural theorem proving. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1910. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/a8901c5e85fb8e1823bbf0f755053672-Abstract-Conference.html.

Sara Rajaee, Kumar Pratik, Gabriele Cesa, and Arash Behboodi. Local look-ahead guidance via verifier-in-the-loop for automated theorem proving. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 16023–16040. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.825. URL https://aclanthology.org/2025.findings-acl.825/.

Cynthia Rudin and Yining Wang. Direct learning to rank and rerank. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 775–783. PMLR, 2018. URL https://proceedings.mlr.press/v84/rudin18a.html.

The mathlib Community. The Lean mathematical library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*. ACM, 2020. doi: 10.1145/3372885.3373824. URL https://doi.org/10.1145/3372885.3373824.

Huajian Xin, Z. Z. Ren, Junxiao Song, Zhihong Shao, Wanjia Zhao, Haocheng Wang, Bo Liu, Liyue Zhang, Xuan Lu, Qiushi Du, Wenjun Gao, Haowei Zhang, Qihao Zhu, Dejian Yang, Zhibin Gou, Z. F. Wu, Fuli Luo, and Chong Ruan. DeepSeek-Prover-V1.5: Harnessing proof assistant feedback for reinforcement learning and monte-carlo tree search. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/b3b55c366d641c07180c40e4f978f311-Abstract-Conference.html.

Kaiyu Yang and Jia Deng. Learning to prove theorems via interacting with proof assistants. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6984–6994. PMLR, 2019. URL https://proceedings.mlr.press/v97/yang19a.html.

Kaiyu Yang, Aidan M. Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan J. Prenger, and Animashree Anandkumar. LeanDojo: Theorem proving with retrieval-augmented language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0944. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/4441469427094f8873d0fecb0c4e1cee-Abstract-Datasets_and_Benchmarks.html. Datasets and Benchmarks Track.

REPRODUCIBILITY STATEMENT

The Supplementary Material gives complete proofs and specifies the scoring and tie rules, theorem splits, checking protocol, outcome definitions, matched evaluation sets, and paired statistical analysis. The source package contains code for analyzing fixed candidate lists, matching corrected outcomes from existing Lean executions, and generating the tables and figure. It identifies the experimental inputs required to repeat these analyses. The package does not run new Lean checks.

AI USE DISCLOSURE

AI tools assisted with manuscript preparation and language polishing. The authors take responsibility for all content.

A COMPLETE PROOFS AND RANKING CONVENTIONS

The results hold on a fixed candidate pool for every family distribution. Fix a state s and a finite nonempty pool C_s of candidate *records*. Each record a has a tactic string $\tau(a)$, a family $f(a)$, a similarity $q_s(a) \in [0,1]$, and a distinct stable index $i(a)$. Two records may contain the same tactic string. Their ranks remain distinct because the experiment retrieves training records without first deduplicating their strings. Acceptance of a record means acceptance of $\tau(a)$ under the specified checker and reconstruction protocol. Throughout a comparison, both the candidate pool and each record's checker outcome remain fixed.

For finite $\lambda \geq 0$, order records lexicographically by

$$\big(q_s(a) + \lambda p_s(f(a)),\ q_s(a),\ -i(a)\big), \tag{11}$$

with larger keys first. Here p_s is a probability distribution on the family labels; a family absent from the predictor receives probability zero. The second component preserves similarity order when perturbed scores tie, and the last component makes every rank unique. This is the convention implemented in the ranking code: sort by the perturbed score and similarity, retaining input order for any remaining tie. Let $a \prec_\lambda b$ mean that a precedes b, so that

$$r_\lambda(a; s) = 1 + \big|\{b \in C_s : b \prec_\lambda a\}\big|.$$

The unperturbed order $\prec_0$ is therefore exactly decreasing similarity with stable index as its tie breaker.

Within-family invariance.

Proof of Theorem 1. If $f(a) = f(b)$, the same constant $\lambda p_s(f(a))$ is added to both similarities. Their perturbed-score difference equals $q_s(a) - q_s(b)$. A strict similarity inequality is consequently preserved for every finite λ. If their similarities are equal, both the first and second components of (11) tie, and their stable indices decide the order independently of λ. Thus the entire ordered subsequence belonging to each family remains unchanged, including all ties. Hard family routing also sorts each family by similarity and then stable index, so it has the same invariance. Only the interleaving of different family subsequences can change. □

Locality of pairwise changes. Write $p_a = p_s(f(a))$ and $q_a = q_s(a)$. The perturbed-score difference is

$$(q_a + \lambda p_a) - (q_b + \lambda p_b) = q_a - q_b + \lambda(p_a - p_b). \tag{12}$$

Since $|p_a - p_b| \leq 1$, a similarity gap larger than λ cannot be reversed. With the similarity tie breaker used here, a nonzero gap equal to λ also cannot be reversed: equality of perturbed scores leaves the larger similarity first. We retain a closed neighborhood in the theorem because it gives a simple, conservative statement that includes similarity ties and does not require endpoint case distinctions.

More precisely, if b follows a in the baseline order, then b passes a exactly when

$$\lambda(p_b - p_a) > q_a - q_b. \tag{13}$$

The inequality is strict even when $q_a = q_b$: equality of the primary scores then preserves the baseline stable-index order. Thus a lower-ranked record at gap exactly λ can be omitted from the possible-crossing set when $\lambda > 0$; at $\lambda = 0$, no comparison changes at all.

For the upper-rank bound, use the set defined in the main text,

$$B^+_\lambda(a; s) = \{b : r_0(b; s) > r_0(a; s),\quad 0 \leq q_a - q_b \leq \lambda\}.$$

For the corresponding lower-rank bound, define

$$B^-_\lambda(a; s) = \{b : r_0(b; s) < r_0(a; s),\quad 0 \leq q_b - q_a \leq \lambda\}.$$

The superscripts refer to the possible increase or decrease of the numerical rank. Both sets can conservatively include records from the same family, although such records cannot change their relative order by Theorem 1.

Proof of Theorem 2. Let D be the records that follow a under $\prec_0$ but precede it under $\prec_\lambda$, and let U be the records that precede a under $\prec_0$ but follow it under $\prec_\lambda$. Counting predecessors gives the exact identity

$$r_\lambda(a;s) - r_0(a;s) = |D| - |U|. \tag{14}$$

Every $b \in D$ has $q_a \geq q_b$. If $q_a - q_b > \lambda$, equation (12) makes the perturbed score of a strictly larger, contradicting $b \prec_\lambda a$. Thus $D \subseteq B^+_\lambda(a;s)$. The symmetric argument gives $U \subseteq B^-_\lambda(a;s)$. Substitution into (14) yields

$$r_0(a;s) - |B^-_\lambda(a;s)| \;\leq\; r_\lambda(a;s) \;\leq\; r_0(a;s) + |B^+_\lambda(a;s)|.$$

This proves both bounds. The same proof allows removal of all same-family records from either neighborhood. For $\lambda = 0$, the ranking keys reproduce $\prec_0$ exactly, even though the closed neighborhoods may contain similarity ties. □

The number of nearby competitors controls rank displacement. A dense similarity interval can contain an arbitrarily large fraction of the pool. For a fixed distribution, the radius can be tightened to $\lambda\Delta_s$, where $\Delta_s = \max_f p_s(f) - \min_f p_s(f) \leq 1$ over the represented families, because the proof uses only a bound on probability differences. This latter radius depends on the supplied distribution; the original radius λ applies uniformly to all distributions.

Exact worst-case rank (Theorem 4). Counting every potentially crossing rival separately can overestimate their simultaneous effect. All records in one family receive the same bonus, and the probabilities assigned to different families must sum to one. The following characterization accounts for both restrictions.

Fix a record a and write $\mathcal{G}_a = \{f(b) : b \in C_s\} \setminus \{f(a)\}$ for the other represented families. For $g \in \mathcal{G}_a$ and $t \geq 0$, define the count

$$N_g(t;a) = \big|\{b \in C_s : r_0(b;s) > r_0(a;s),\ f(b) = g,\ q_a - q_b < t\}\big|. \tag{15}$$

In particular, $N_g(0;a) = 0$. The strict inequality incorporates the tie rule in Equation (13).

Let $r_{\lambda,p}(a;s)$ denote the rank under weight $\lambda \geq 0$ and family distribution p. If $\mathcal{G}_a$ is nonempty, then

$$\max_p r_{\lambda,p}(a;s) = r_0(a;s) + \max_{\substack{x_g \geq 0\ (g\in\mathcal{G}_a) \\ \sum_{g\in\mathcal{G}_a} x_g \leq 1}} \sum_{g\in\mathcal{G}_a} N_g(\lambda x_g; a). \tag{16}$$

The left maximum ranges over all probability distributions on the represented families; allowing probability on additional, unrepresented families leaves its value unchanged. If $\mathcal{G}_a$ is empty, the rank is $r_0(a;s)$ for every distribution.

Proof of Theorem 4. Suppose $\mathcal{G}_a \neq \varnothing$. Starting from any p, set the probability of $f(a)$ to zero and move its mass, together with any mass on unrepresented families, to one family in $\mathcal{G}_a$. Every other represented family's probability is unchanged or increased. For every $g \in \mathcal{G}_a$, its probability difference relative to $f(a)$ is therefore at least its original value. All cross-family comparisons can only move against a, while within-family comparisons remain fixed. Consequently, some rank-maximizing distribution assigns zero probability to $f(a)$ and total probability one to $\mathcal{G}_a$.

Under such a distribution, every baseline predecessor of a remains before it. A baseline successor in family g passes a exactly when its gap is strictly less than $\lambda p(g)$, by Equation (13). The rank is consequently $r_0(a;s) + \sum_g N_g(\lambda p(g); a)$. This proves Equation (16) with $\sum_g x_g = 1$. Allowing $\sum_g x_g \leq 1$ does not change the maximum: any unused mass can be added to one represented rival family, and every count is nondecreasing in that family's mass. Although these count functions have jumps, their joint image is a finite nonempty set of integers, so a maximum value is attained. If no rival family is represented, within-family invariance proves the final assertion. □

There is also a finite form of this maximum. For each $g \in \mathcal{G}_a$, let ℓ_g be the number of baseline successors of a in family g, and list their nonnegative similarity gaps in nondecreasing order as $d_g(1), \ldots, d_g(\ell_g)$. Set $d_g(0) = 0$. For $\lambda > 0$, Equation (16) is equivalent to

$$\max_p r_{\lambda,p}(a;s) = r_0(a;s) + \max\left\{ \sum_{g\in\mathcal{G}_a} m_g : m_g \in \{0,\ldots,\ell_g\},\quad \sum_{g\in\mathcal{G}_a} d_g(m_g) < \lambda \right\}. \tag{17}$$

Indeed, passing at least $m_g > 0$ records from family g requires $x_g > d_g(m_g)/\lambda$. These strict inequalities can hold simultaneously exactly when their thresholds sum to less than one; any positive remaining mass can be distributed to make them all strict. Choosing all $m_g = 0$ is feasible. At $\lambda = 0$, the rank is unchanged and the finite form is replaced by this identity.

For an accepted witness a, requiring the right-hand side of Equation (16) to be at most k is necessary and sufficient for *that same record* to remain in the top k under every family distribution. It is only sufficient for exposure of the accepted set: different accepted records may protect exposure under different distributions.

For example, take three records in three families, with similarities $q_a = 0.80$ and $q_b = q_c = 0.65$, and let $\lambda = 0.25$. Record a is initially first. Each rival can pass it under a point-mass prediction, so the separate-rival bound permits rank three. Simultaneous passage would require $p(f(b)) - p(f(a)) > 0.60$ and $p(f(c)) - p(f(a)) > 0.60$, which is impossible for a probability distribution. Equation (16) gives worst-case rank two, attained by assigning probability one to either rival family. If a is accepted, it therefore remains exposed at cutoff two for every prediction, although the unrefined similarity-neighborhood condition fails.

Exposure preservation. Fix a checker environment E and abbreviate $A_s = A_E(s) \subseteq C_s$. For an integer cutoff $1 \le k \le |C_s|$, write

$$E_\lambda(s; k) = \mathbf{1}\{\exists a \in A_s : r_\lambda(a; s) \le k\}.$$

Proof of Corollary 3. Suppose an accepted record a satisfies

$$r_0(a; s) + |B^+_\lambda(a; s)| \le k.$$

Theorem 2 gives $r_\lambda(a; s) \le k$, so the same accepted record remains exposed. The condition depends only on the unperturbed ranking, similarities, the chosen weight, and acceptance of this witness. It therefore works for every possible family distribution, including an incorrect point-mass prediction. If several accepted records satisfy it, any one is sufficient. No condition is imposed on the ranks of the remaining accepted records. □

An alternative sufficient condition uses the top-k similarity margin. If an accepted a lies in the baseline top k, and its similarity exceeds every baseline record outside that top k by more than λ, then no outside record can pass a. At most $k - 1$ records can precede it after reranking, so it remains exposed. Averaging the indicator of the displayed sufficient condition over states supplies a deterministic lower bound on aggregate exposure, not an estimate of the actual acceptance rate.

B HARD ROUTING, LIMITING ORDERS, AND TIGHT EXAMPLES

Hard routing uses a total order ρ_s of families supplied by the classifier, breaking equal family probabilities by the classifier's stable class order. It concatenates the family blocks in that order and retains the baseline order inside each block. Define

$$n_{<f}(s) = \sum_{g:\rho_s(g)<\rho_s(f)} |\{b \in C_s : f(b) = g\}|,$$

and let $j_f(a; s)$ be the rank of a within its own family under decreasing similarity and stable index.

Proof of Proposition 5. Every record in a preceding family is placed before a, contributing $n_{<f(a)}(s)$ predecessors. Exactly $j_{f(a)}(a; s) - 1$ records of its own family precede it, and no record from a later family does. Consequently,

$$r_{\mathrm{hard}}(a; s) = n_{<f(a)}(s) + j_{f(a)}(a; s).$$

In particular, if the first predicted family contains m records and excludes a, then $r_{\mathrm{hard}}(a; s) \ge m+1$, irrespective of the similarity gap. To see that this can destroy exposure, take one accepted record a with similarity 1 and family F, and m rejected records with similarity 0 and family G. Predict G before F. Baseline retrieval ranks a first; hard routing ranks it $m + 1$. For every $k \le m$, exposure changes from one to zero. Since m is arbitrary, no pool-size-independent rank bound, or protection based on a similarity neighborhood smaller than the full score range, is available for this hard gate. □

What the infinite-weight endpoint means. For a finite pool with pairwise distinct family probabilities, the soft order eventually equals hard routing as λ increases. If at least two families are represented, let $\delta > 0$ be their minimum probability difference. Since similarities lie in $[0, 1]$, every $\lambda > 1/\delta$ makes all inter-family comparisons agree with decreasing family probability; within-family order is already invariant. With only one represented family, the orders coincide for every weight. This proves the endpoint interpretation without assuming anything about prediction accuracy.

Probability ties require an explicit qualification. The literal large-λ limit of (11) orders tied-probability families together by similarity. The implemented hard ranker instead resolves the family tie first and then concatenates the corresponding blocks. Therefore the hard endpoint is defined with this additional family-order refinement; it is the literal limit when represented family probabilities are distinct. Both versions preserve within-family order, and the hard-routing counterexample uses strict family preference, so none of the preceding bounds or failure examples relies on this distinction.

Tightness and the limits of soft protection. Fix any $\lambda > 0$ and choose $0 < \varepsilon < \min\{\lambda, 1\}$. Let a have similarity 1 and probability $p_s(F) = 0$, and let m records of another family G have similarity $1 - \varepsilon$ and $p_s(G) = 1$. Then a starts at rank one, all m rivals belong to $B^+_\lambda(a; s)$, and every rival obtains the larger perturbed score $1 - \varepsilon + \lambda$. Thus

$$r_\lambda(a; s) = m + 1 = r_0(a; s) + |B^+_\lambda(a; s)|.$$

The upper bound is attained. If a alone is accepted, every cutoff $k \leq m$ loses exposure despite the soft prior. For the upward example, assign a similarity $1 - \varepsilon$ and family probability one, and assign m rivals similarity 1 and family probability zero. Its rank then moves from $m + 1$ to one, attaining the lower bound. Both extremes arise within one similarity interval. They show how densely packed records turn a small score change into a large rank change.

C THE FIXED-LIST CEILING

Proof of Proposition 6. Let $L_s \subseteq C_s$ be a fixed list of ℓ candidate records, and let a reranker output any permutation σ_s of this list. For $1 \leq k \leq \ell$, its top-k set is a subset of L_s, so

$$\mathbf{1}\{\mathrm{Top}_k(\sigma_s) \cap A_s \neq \varnothing\} \leq \mathbf{1}\{L_s \cap A_s \neq \varnothing\}.$$

At $k = \ell$, equality holds for every permutation. For any positive cutoff, an oracle ordering attains this upper bound by placing an accepted record first whenever one exists. Thus the fixed list determines attainable exposure, while reranking determines how early its accepted records appear. The statement is pointwise and therefore remains true after averaging over any collection of states. □

The ceiling applies to permutations of fixed records with fixed checker outcomes. Tactic repair, argument synthesis, adaptive generation, or changes to the checking environment can alter the candidate set or its accepted subset. Those operations change the selection problem to which the ceiling applies.

D DATA AND RANKING PROTOCOLS

D.1 PROOF STATES AND CANDIDATE RECORDS

The dataset comprises 3,723 proof steps from 1,702 theorems in 46 mathlib4 files, with 81 tactic families and a mean of 2.187 recorded steps per theorem. The files were selected from arithmetic, mixed-library, and proof-dense modules. This curated selection yields a collection dominated by rewriting, simplification, and direct proof terms. LeanDojo extracts the traces, and Lean evaluates candidate tactics (Yang et al., 2023; de Moura & Ullrich, 2021; The mathlib Community, 2020).

Each record contains a source file, theorem identifier, step index, pretty-printed main goal, local context, recorded next tactic, and tactic family. The family label records the outer tactic form; the candidate must also supply its arguments, including rewrite lemmas, local identifiers, and proof terms. Training records form the candidate pool, and several records may contain the same tactic string. Each ranker returns five record positions. The number of distinct tactics in those positions depends on the ranking.

Similarity retrieval and learned rankers use state-only inputs. The representation excludes the recorded next tactic, post-state, annotated tactic syntax, and premises used by the next tactic. Separate representation experiments added file and theorem metadata, retrieved premises, or future-premise annotations; those extensions are absent here. Future-premise annotations reveal the recorded continuation and are unavailable when selecting a next action. The trace-family oracle, defined below, receives the recorded family explicitly.

D.2 Similarity, family prediction, and additional rankers

Term frequency–inverse document frequency (TF–IDF) vectors represent training and query states. The transform uses whitespace tokenization, preserves case, sets the minimum document frequency to one, and is fitted on training states. Cosine similarity supplies the baseline score. A TF–IDF logistic-regression classifier estimates family probabilities using the same tokenization, balanced class weights, and a maximum of 1,000 iterations. Families are ordered by decreasing probability.

Equation (11) defines the soft score and its tie rule. Hard routing concatenates the similarity-ordered blocks of all predicted families. When the first family contains at least five records, it occupies the entire top five; off-family records follow that block. Families absent from the predictor follow all predicted families.

The reciprocal-rank-fusion (RRF) score is

$$\frac{1}{60 + r_0(a; s)} + \frac{1}{60 + \rho_s(f(a))},$$

where $r_0(a; s)$ is the baseline rank of record a at state s, and $\rho_s(f(a))$ is its family's predicted rank; both are one-based. Ties are resolved by similarity and stable record order. Top-three interleaving sorts records within the three highest-priority families, then takes one record from each family in repeated rounds, skipping exhausted families. Records from the remaining families follow in similarity order. These two rules fall outside the additive probability-score class of the soft rank-displacement theorem.

The trace-family oracle places the recorded next tactic's family first and preserves similarity order within every family. It measures the effect of supplying the trace family while keeping within-family retrieval fixed. Its exact@5 depends on whether the recorded tactic is in the pool and where it ranks within its family. Since acceptable alternatives can belong to other families, oracle acceptance is not an upper bound over ranking rules.

D.3 Trace evaluation and execution evaluation

The six-split trace study partitions theorem identifiers into 60% training, 10% validation, and 30% test groups, using split identifiers 0, 1, 2, 3, 4, and 42. Sorted theorem identifiers are shuffled and partitioned using rounded group counts. For split 42, the groups contain 1,021, 170, and 511 theorems, corresponding to 2,183, 351, and 1,189 step records. Each theorem belongs to one partition. Different theorems from the same source file may occur in different partitions. Normalizing state text gives 3,572 distinct states, with 146 duplicate groups containing 297 records. Identical normalized state text can therefore cross theorem partitions.

The soft weight is selected from $\{0, 0.05, 0.2, 0.5, 1\}$ using validation exact@5. Ties are resolved, in order, by validation exact@1, family@5, and the smaller weight. The split-42 selection is 0.2. The test candidate pool contains training records only. Table 6 gives the variation over the six splits. Its exact-match entries compare stored tactic strings directly; trace-hit events on the corrected execution panel use whitespace-collapsed strings. The first-family rank is averaged over states whose recorded family occurs in the training pool.

The execution study uses a separate 70%/30% theorem split with split identifier 42, giving 2,534 training records and 1,189 test records. A deterministic shuffle selects 500 test states from 281 theorems. Five rankers each supply five candidate records per state: 12,500 positions in total, representing 4,775 unique state–tactic pairs after whitespace normalization. All candidate identities and positions remain fixed in the corrected comparison. The execution and six-split trace studies use different training pools and soft-weight selections and are reported separately.

The original execution file does not store the soft weight as a field. Reconstructing its state-only ranking with the documented executor default $\lambda = 0.25$ reproduces the exact tactic string and family

Table 6: Variation across six theorem splits. Entries are mean ± population standard deviation, not confidence intervals.

Ranker	Family@5	Exact@5	First family rank
Unguided	0.600 ± 0.016	0.169 ± 0.012	40.8 ± 9.9
Soft family	0.589 ± 0.020	0.170 ± 0.011	42.4 ± 10.1
Hard family	0.130 ± 0.015	0.082 ± 0.008	531.3 ± 28.3
Family RRF	0.546 ± 0.018	0.169 ± 0.010	94.6 ± 22.0
Top-three interleaving	0.231 ± 0.009	0.111 ± 0.005	102.9 ± 8.0
Trace-family oracle	0.988 ± 0.006	0.245 ± 0.021	1.0 ± 0.0

in all 2,500 frozen soft positions. The validation-selected value $\lambda = 0.2$ matches only 2,382 positions. We therefore describe the execution lists as exactly compatible with the fixed-weight 0.25 rule, without claiming that a finite ranking uniquely identifies a numerical weight.

E Corrected checker outcomes

E.1 Line attribution and target exclusion

The original execution labels were affected by incorrect line attribution and exposure of the target theorem. In batched Lean files, the recorded source spans counted each tactic as one line. Every additional line in a multiline tactic shifted subsequent candidates relative to Lean's physical error-line locations, assigning some rejections to the wrong candidates. Importing the completed module also made the target theorem and its attributes available during checking. A tactic could invoke the target directly or use it through automation.

Among the 4,775 unique historical state–tactic pairs, 347 receive conflicting acceptance labels across repeated appearances. We replace the historical labels for all matched candidates. The corrected results supersede the previously reported Accept@5 values of 0.672, 0.652, and 0.580 for unguided, soft, and hard ranking, respectively, and the 53.4% acceptance-without-trace rate. Candidate identities and ranks were determined before checking and are unchanged.

The corrected checker aligns candidate spans with physical Lean lines and uses candidate-specific markers to identify resulting states. It reconstructs the source preceding the target declaration, resolves the source namespace, and records tactic rejection separately from failure to reconstruct the checking environment. Corrected replay outcomes are available for both this target-excluding prefix and the completed module. The difference between these replay conditions measures sensitivity to the checking environment. Historical-to-corrected differences also reflect line-mapping and namespace corrections and cannot isolate target exposure.

E.2 Acceptance and closure

The checker records the state immediately after the candidate tactic. Acceptance means execution without a tactic error; the tactic may close the goal, create new goals, or leave the state unchanged. Closure requires zero remaining goals at the same observation point. Reconstructed examples with residual goals may end with a trailing `sorry`, inserted after the outcome is recorded and excluded from the candidate and closure assessment. None of the retrieved candidates in the original 500-state lists contains `sorry` or `admit`.

Each metric counts states whose list contains at least one candidate satisfying its criterion. On the primary 497-state panel, unguided retrieval has Accept@5 of $205/497$ and Close@5 of $77/497$. The corresponding counts are 203 and 78 for soft ranking, and 170 and 53 for hard routing.

E.3 Matching fixed lists to corrected outcomes

Each candidate is matched to a corrected outcome by its state identifier and whitespace-collapsed tactic string, following the convention of the replay records. Whitespace normalization serves only to match these records; arbitrary changes to Lean source whitespace can alter a program's meaning. The

corrected outcome table contains exactly one row per pair. Repeated candidates retain their original ranking positions and receive the same matched outcome.

All 2,500 candidate positions are matched for unguided, soft, and RRF ranking. Hard routing has 2,498 matched positions, with outcomes missing for two pairs in one state. Top-three interleaving has 2,494 matched positions, with one pair missing in each of six states. The primary panel requires five matched positions for each of unguided, soft, and hard ranking. It also excludes a state if any required pair has an environment reconstruction error in any of the stored replay conditions. This rule leaves 497 states from 278 theorems and 30 source files. One state is excluded for missing hard-routing outcomes and two for reconstruction errors.

Applying the same inclusion rule to all five rankers gives a secondary panel of 491 states from 275 theorems and 30 files. A state with five validly recorded rejections for every ranker remains eligible. Missing outcomes remain unobserved and are excluded by the matching rule. Section F bounds the effect of the three primary-panel exclusions.

F Paired inference and supplementary results

F.1 Paired differences and clustered intervals

For each retained state s, let d_s denote the competing ranker's binary top-five exposure minus unguided exposure. The estimated difference is $\widehat{\Delta} = N^{-1} \sum_s d_s$, where N is the number of retained states. Each bootstrap replicate samples theorem groups with replacement, includes every retained state of each selected theorem with its sampled multiplicity, and recomputes the state-weighted mean. A replicate contains as many theorem draws as there are distinct theorems in the panel. Its number of states can vary with the sampled theorem sizes. We use 5,000 replicates with random-state identifier 20270926. The 95% interval endpoints are entries 125 and 4,875 in the sorted replicate means, using lower-index 2.5th and 97.5th percentiles. The intervals are retrospective and apply to individual comparisons without adjustment for simultaneous inference.

On the primary panel, hard routing gains exposure on 35 states and loses it on 70. Its mean difference is $-35/497$, or -7.04 percentage points, with a theorem-cluster interval of $[-11.80, -2.59]$ points. Soft ranking gains on four states and loses on six, giving $-2/497$, or -0.40 points, with interval $[-1.70, 0.84]$. Resampling individual states instead gives intervals of $[-11.27, -3.02]$ and $[-1.61, 0.80]$ points, respectively. The main text reports the theorem-cluster intervals.

The exact two-sided McNemar test conditions on the number of discordant states, with each direction assigned probability one half under the null. It gives $p = 0.000821$ for hard routing and $p = 0.754$ for soft ranking under an independent-state model. The main text uses theorem-cluster intervals to account for dependence within theorems. The resampling treats different theorems as independent, including those in one source file, and estimates uncertainty on the curated collection.

F.2 Bounds for excluded states

The primary comparison omits three of the original 500 states. Allowing each missing state an exposure difference anywhere in $[-1, 1]$ gives full-panel hard-minus-unguided bounds of $(-35 - 3)/500$ and $(-35 + 3)/500$, or $[-7.6, -6.4]$ points. The corresponding soft-minus-unguided interval is $[-1.0, 0.2]$ points. These deterministic missing-outcome bounds are distinct from confidence intervals. Every possible resolution of the three exclusions retains a negative hard-routing difference; the soft difference can have either sign.

The analogous 500-state acceptance bounds are $[205/500, 208/500] = [0.410, 0.416]$ for unguided, $[203/500, 206/500] = [0.406, 0.412]$ for soft, and $[170/500, 173/500] = [0.340, 0.346]$ for hard routing. They allow either outcome for each excluded state, including states with partially observed candidates.

F.3 Accepted alternatives on identical states

Table 7 counts states with a trace hit, an accepted action, an accepted non-trace action, acceptance without a trace hit, and a trace hit without acceptance. A list may contain both the recorded tactic

Table 7: Different top-five events on the same 497 target-excluding states. Non-trace means at least one accepted non-trace action; A without T requires no trace hit anywhere in the list.

Ranker	Trace (T)	Accept (A)	Non-trace	A without T	T without A
Unguided	83	205	164	139	17
Soft family	83	203	158	138	18
Hard family	44	170	137	132	6

Table 8: All five rankers on 491 common states (275 theorems), using repaired target-excluding outcomes. Differences and individual 95% theorem-cluster intervals are in percentage points; these secondary comparisons have no simultaneous adjustment.

Ranker	Hits@5	Accept@5	Δ (pp)	95% CI
Unguided	202	0.411	–	–
Soft family	201	0.409	-0.20	[-1.37, +0.99]
Hard family	168	0.342	-6.92	[-11.68, -2.25]
Family RRF	206	0.420	+0.81	[-1.67, +3.35]
Top-three interleaving	224	0.456	+4.48	[+0.65, +8.37]

and an accepted alternative, so these events overlap. Accept@5 minus exact@5 equals the rate of acceptance without a trace hit minus the rate of a trace hit without acceptance. Recovering either rate separately requires the joint counts.

For unguided retrieval, 66 states have both a trace hit and an accepted list: $205 - 139 = 83 - 17 = 66$. The corresponding two-by-two counts are 66 with both events, 139 with acceptance only, 17 with a trace hit only, and 275 with neither, summing to 497. Of the 164 states with an accepted non-trace candidate, 139 have no trace hit and 25 have a trace hit elsewhere in the list. These 25 states account for the difference between accepted non-trace actions and acceptance without a trace hit.

F.4 Additional rankers and cutoff dependence

On the 491-state shared panel, top-three interleaving has the highest Accept@5 point estimate, $224/491 \approx 0.456$, compared with $202/491 \approx 0.411$ for unguided retrieval. Its paired difference is $+4.48$ percentage points, with a theorem-cluster interval of $[0.65, 8.37]$ points. RRF changes exposure by $+0.81$ points with interval $[-1.67, 3.35]$. Interleaving improves measured exposure on this panel; RRF's interval spans zero.

The ordering changes with the cutoff. On the primary panel, hard routing has Accept@1 of $143/497 \approx 0.288$, compared with $111/497 \approx 0.223$ for unguided and $119/497 \approx 0.239$ for soft ranking. At cutoff three, the respective acceptance counts are 159, 173, and 175. Hard routing leads at the first position and falls behind by the third. Its first-action advantage is outweighed by the additional acceptable candidates exposed by the other lists.

Acceptance and closure favor different methods on the all-five panel. Top-three interleaving exposes a closing action in 61 states, compared with 76 for unguided retrieval, despite its larger acceptance count. It provides executable next moves on more states and immediately closing moves on fewer. Evaluating the effect on proof search requires a continuation policy and theorem-level outcomes.

F.5 Sensitivity to the checking environment

For the same 497 states and lists, corrected completed-module checking accepts at least one candidate in 221 states for unguided, 219 for soft, and 175 for hard routing. The target-excluding counts are 205, 203, and 170, giving net differences of 16, 16, and five states. Both conditions use corrected line attribution and fixed candidate lists. Their difference measures sensitivity to the checking context. The completed module makes the target and later declarations available, so an accepted action in that environment can depend on the theorem being proved.

Table 9: Matched repaired replay on 497 states. E_0 excludes the target; E_3 imports the completed module. Both columns use repaired label attribution. Closure requires no remaining goals after the candidate, before administrative completion.

Ranker	E_0 Accept@5	E_3 Accept@5	E_0 Close@5	E_3 Close@5
Unguided	0.412	0.445	0.155	0.201
Soft family	0.408	0.441	0.157	0.203
Hard family	0.342	0.352	0.107	0.117

G SCOPE AND REPRODUCIBILITY

The analysis uses fixed candidate lists and corrected outcomes from existing Lean executions, with no new Lean executions. The stored lists support direct comparisons of their exposure. Measuring how often the sufficient exposure condition holds additionally requires complete per-state similarity neighborhoods, which the stored short lists omit.

The analysis code constructs the matched evaluation sets, verifies five-record lists and unique outcome matching, and computes paired event counts. It reports acceptance and closure at cutoffs one, three, and five; normalized trace hits; accepted non-trace actions; gains and losses relative to unguided retrieval; and state and theorem-cluster intervals. The analysis materials include input-file digests, excluded-state details, and regeneration instructions. All numerical tables are generated from these summaries.

The corrected executions use Lean 4.28.0. The available timing records refer to the earlier checker, so execution costs are not compared here.

Relationship to the workshop study. The earlier workshop study (Kamat & Wang, 2026) introduced the collection, ranker family, trace comparisons, and pairwise gap intuition. This paper derives rank-displacement and exposure conditions, characterizes the exact worst-case rank under shared family probabilities, specifies how probability ties affect the hard-routing limit, and reassesses fixed lists using corrected matched outcomes. Later context-augmentation experiments study how labels from altered checking environments affect learned scorers. Those experiments use larger classifiers, proof search, and pooled classification measures; the comparisons here hold the candidate lists fixed.

Open empirical questions. Candidate deduplication, local-name repair, and calibration-based weight selection provide ways to test the effects of repeated strings, tactic portability, and family-probability quality. Language-model generation, unseen-project transfer, and end-to-end proving remain open empirical questions. The exposure condition concerns preservation of an accepted witness already in the pool; Lean execution determines which candidates supply such a witness.