BRIDGEBANK: PREDICTING REUSE BEFORE ADMITTING VERIFIED LEAN REPAIRS

Anonymous authors
Paper under double-blind review

ABSTRACT

A solve-only comparison favors disabling novelty filtering over random reuse ranking on miniF2F solve (41.4% versus 41.1%), while the ordering reverses on observed Reuse@12 (6.2% versus 9.1%) and fully exposed unused admissions per 100 policy-specific failures (63.4 versus 14.2). This inversion exposes the decision left open by verification: after several repairs type-check, which one, if any, should enter shared retrieval state? BRIDGEBANK trains a top-one-or-none admission score from completed 12-round future-reference cohorts; only typed, retrieval-distinct repairs from diagnosed missing-bridge failures are eligible. Across family-held-out miniF2F and Mathlib-Structured runs capped at 576 A100-hours, it improves Reuse@12 over broad post-failure growth by $+7.1$ $[5.5, 8.7]$ and $+7.5$ $[5.9, 9.1]$ points while the solve differences are $+0.5$ $[-0.1, 1.1]$ and $+0.7$ $[0.0, 1.4]$. The resulting libraries contain 520 rather than 1,833 fully exposed but unused miniF2F admissions and 843 rather than 3,488 on Mathlib-Structured. Across recurrence strata, the paired reuse advantage rises from $+2.6$ $[1.0, 4.2]$ to $+11.5$ $[9.1, 13.9]$ points. Together, these comparisons identify predicted-reference-guided admission as a route to higher-reuse, lower-clutter libraries at comparable solve in the evaluated regime.

1 INTRODUCTION

On miniF2F, disabling novelty filtering ranks 0.3 points above random reuse ranking by solve. On the two reported library-facing coordinates, random ranking instead has higher observed reuse and 77.6% fewer unused admissions per 100 failures. The disagreement is not a checker failure—every admitted statement has already passed Lean. It arises because proof completion and library mutation answer different questions.

The distinction becomes consequential whenever a prover carries generated lemmas into later searches (Wang et al., 2024; Yang et al., 2023; Xin et al., 2025). A verified helper joins the premise index, competes with established declarations, and changes the context seen by subsequent goals. We therefore formulate *delayed-utility admission*: after a failed trace yields verified repairs, the policy keeps at most one or leaves shared state unchanged. Its outcome has five visible coordinates—future solve, observed downstream reference, proof shortening on reused proofs, fully exposed but unused objects, and end-to-end compute—rather than a single type-check bit.

BRIDGEBANK realizes this policy as temporally supervised selection. A calibrated failure model opens generation only for missing-bridge traces; typed repair operators produce at most 40 candidates; an adversarial novelty test removes statements already represented in the accessible library; and a ranker trained on completed 12-round cohorts predicts whether a survivor will be referenced later. The ranker and every threshold are frozen before test families are processed, and its no-admit threshold makes an unchanged library a first-class outcome.

The end-to-end contrast uses the same goal order, generator, family partitions, follow-up horizon, and 576-A100-hour cap for BRIDGEBANK and BroadGrow-PF, a LEGO-Prover-style broad-retention policy (Wang et al., 2024). Their solve rates remain close, but the induced libraries diverge: the family-blocked reuse advantage is 7.1–7.5 points, and BRIDGEBANK leaves 520 instead of 1,833 unused admissions on miniF2F and 843 instead of 3,488 on Mathlib-Structured.

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.

Contributions. First, we separate checked repair from persistent admission and formalize the solve–persistence rank inversion that a solve-only comparison can miss. Second, BRIDGEBANK couples failure-conditioned typed generation to completed-cohort future-reference supervision and an explicit null action. Third, the family-blocked broad-growth contrast establishes the library-level reuse/clutter separation, near-volume interventions provide descriptive evidence about ranker behavior, and recurrence strata locate the largest policy differences. The result is a measurable admission problem for formal systems that learn from their own verified repairs.

2 POST-VERIFICATION ADMISSION IN GROWING PROVERS

Proof generation systems improve the current theorem through language-model pretraining, tactic-state interaction, retrieval, tree search, reinforcement learning, and ATP integration (Polu & Sutskever, 2020; Polu et al., 2023; Han et al., 2022; Lample et al., 2022; Jiang et al., 2022; Yang et al., 2023; Xin et al., 2025; Wu et al., 2021; Hubert et al., 2026). Whole-proof repair revises a candidate proof (First et al., 2023); premise selection and hammers alter its search context (Irving et al., 2016; Loos et al., 2017; Jiang et al., 2022); and completed traces can train later tactic or premise choices (Gauthier et al., 2021). These methods provide the proving machinery used here. BRIDGEBANK begins at a later interface, when checking has produced an eligible set and the remaining operation is a persistent index update.

Growing-library systems make verified outputs available beyond one goal, but their central control variables differ. LEGO-Prover creates and evolves skills during modular proving (Wang et al., 2024); DreamProver alternates lemma production with wake–sleep consolidation (Zhang et al., 2026); LeanAgent organizes curriculum and retriever updates across repositories (Kumarappan et al., 2025); the Conjecturing–Proving Loop conditions discovery on earlier theorems (Kasaura et al., 2025); and MathlibLemma targets reusable folklore-statement generation (Liu et al., 2026). BRIDGEBANK differs at a specific decision interface: it acts only on verified candidates produced from a failed trace, trains on completed 12-round future-reference cohorts, permits a null mutation, and enforces a one-object cap. Because the external systems use different generated units, provers, corpora, and update schedules, our controlled evidence compares the in-stack BroadGrow-PF and ablation policies rather than their published headline scores.

Testing-based theorem assessment asks whether successor theorems continue to compile when a generated statement is substituted (Kim et al., 2026), while Mathlib maintenance makes naming, redundancy, integration cost, and review burden explicit library concerns (Baanen et al., 2026). We instead measure the policy-induced library after admission: observed future references, positive-only shortening on qualifying reruns, direct unused counts, solve, and compute. Failure routing draws on refinement and program repair (Clarke et al., 2003; Le Goues et al., 2019), and typed proposal generation follows formal-reasoning systems (Jiang et al., 2023; Wu et al., 2022; Azerbayev et al., 2024); the distinctive learning signal is supplied only after a candidate's full future-exposure cohort has matured.

Table 1 locates the decision precisely. Prompt repair and retrieval/search terminate at the current theorem. B2 retains a verified mutation used by its repair, B3 retains mined helpers, and B7 retains a batch of novelty survivors; their keep events are tied to immediate proof support or verification. BRIDGEBANK can abstain before generation and reject every verified proposal on predicted future reference. The random-rank and immediate-supervision controls later hold its trigger, candidate set, novelty rule, and one-object cap fixed.

3 THE POST-VERIFICATION PERSISTENCE DECISION

Let $P_\theta(g \mid \mathcal{L})$ be a prover solving goal g with persistent library $\mathcal{L}$. A failed attempt emits trace $\tau = (\mathbf{s}, \mathbf{t}, \mathbf{p}, \mathbf{m}, b)$: proof states, tactics, premises, metavariables, and consumed budget. Generation under budget B_{gen} produces proposals, and Lean checking yields the verified set $V(\tau)$. Persistence is a separate action $A \subseteq V(\tau)$ that mutates $\mathcal{L}$ before future goals $\mathcal{G}_H$ arrive.

Definition 1 (Delayed utility of persistent repair)**.** *For $A \subseteq V(\tau)$ with $|A| \le K$, let $\Delta S_H(A)$ be future solve improvement relative to an operator-specified reference, $R_H(A)$ the observed fraction of admissions referenced at least once, $Q_H(A)$ the mean $100(s_{\text{ref}} - s_A)/s_{\text{ref}}$ over all paired reruns*

where both proofs succeed (including ties and lengthenings), $C_H(A)$ the fully exposed accepted-but-unused burden, and $E_H(A)$ end-to-end compute. The persistence objective is

$$A^\star = \arg\max_{|A|\le K} \mathbb{E}[\alpha\Delta S_H(A) + \beta R_H(A) + \eta Q_H(A) - \lambda C_H(A) - \mu E_H(A) \mid \tau]\,. \tag{1}$$

The empty action is feasible; verification grants eligibility, and the admission policy determines persistence.

We write $\mathcal{U}_H(A \mid \tau)$ for the conditional expectation in Equation 1, and omit the conditioning argument when the context is clear.

All coefficients are nonnegative and express an operator's library policy. Equation 1 is a decision specification, not the label fitted by BRIDGEBANK: the ranker predicts observed future reference, while solve, signed proof-step change, unused burden, and compute remain separate policy-level outcomes. The reported Short.$^+$ is a positive-only conditional diagnostic, not an estimate of Q_H. We report coordinates rather than claim an unidentified scalarization or interpret a policy-induced reference as a lemma-level causal effect.

Proposition 1 (Solve–persistence rank inversion)**.** *For policies a and b, write $D_j = \beta R_j + \eta Q_j - \lambda C_j - \mu E_j$. If $\Delta S_a > \Delta S_b$ but $D_b - D_a > \alpha(\Delta S_a - \Delta S_b)$, solve ranks a above b while delayed utility ranks b above a.*

The proposition follows by subtracting the two objectives; Appendix G gives the algebra. It shows why solve cannot determine a persistence ordering once future reference and library cost enter the decision. Section 5 reports a concrete disagreement among solve, observed reuse, and unused burden; without selecting the coefficients, we treat that disagreement as an ordering diagnostic rather than a numerical estimate of $\mathcal{U}_H$.

Metric semantics. Reuse@12 is the fraction of accepted lemmas referenced by at least one successful proof during rounds $r+1, \ldots, r+12$ after admission at round r. It is an observed policy-trajectory quantity: the originating proof is excluded, repeated uses of one lemma count once in the numerator, and exposure-only follow-up rounds complete late cohorts, so no accepted item is right-censored. Unused/100 failures divides a policy's fully exposed zero-use admissions by that same policy's failed traces; it is used only where row-specific denominators are displayed. When both admission-enabled and masked reruns succeed and the enabled proof is shorter, Short.$^+$ reports $100(s_{\text{mask}} - s_{\text{admit}})/s_{\text{mask}}$; this conditional diagnostic excludes failed, tied, and lengthened reruns and is not an estimator of Q_H. Solve is the fraction of distinct test goals solved by the end of the 12-round induction stream. Solve, reuse, and shortening are computed within seed and then averaged; direct A/R/U counts pool the three seeds and therefore need not reconstruct rounded seed-level means.

Table 1: Persistence-boundary comparison. B7 supplies broad growth with the same base prover, generator, goal order, family holdout, exposure horizon, and end-to-end compute cap; every policy retains its own failure stream.

Line	System action	Payoff horizon	Keep rule	Evidence here
Broad growth	Add a lemma batch	Later goals	Retain verified novelty survivors up to cap	B7, shared stack
Controlled mining	Extract a helper	Later goals	Completed-trace signal	B3, shared cap
Repair + retention	Keep a used mutation	Current and later goals	Verified repair support	B2, shared cap
Prompt repair	Revise current proof	Current goal	No persistent update	B1
Retrieval/search	Expand current search	Current goal	No persistent update	B4–B6
BRIDGEBANK	Keep one repair or none	12 future rounds	Predicted future reference	Full protocol

4 BRIDGEBANK: A CONSTRAINED DELAYED-CREDIT GATE

BRIDGEBANK is a constrained surrogate for Eq. 1 spanning failure routing through the interface between Lean checking and library mutation. Diagnosis supplies an early null action, typed operators construct the feasible set, novelty filtering removes retrieval-equivalent candidates, and temporal ranking estimates the probability of a future reference. A development-fixed threshold converts the top score into a keep/discard decision. The method therefore learns one delayed coordinate under an explicit top-one constraint; solve, shortening, clutter, and compute evaluate the resulting policy rather than serving as unobserved training labels.

Failure diagnosis. A calibrated six-layer transformer D_ϕ with hidden size 512 predicts missing auxiliary lemma, retrieval failure, underpowered search, malformed decomposition, or false target from the trace. Its focal objective includes an ECE penalty (Lin et al., 2017):

$$\mathcal{L}_{\text{diag}} = \mathcal{L}_{\text{focal}}(D_\phi(\tau), y) + \lambda_{\text{ece}}\mathcal{L}_{\text{ECE}}(D_\phi), \tag{2}$$

with $\lambda_{\text{ece}} = 0.3$. Training uses 3,200 two-annotator real traces ($\kappa = 0.79$) and 8,500 injected failures. A missing-bridge probability above $\gamma = 0.4$ activates generation. This routing gate is fixed on Synthetic-Gap development data; the 0.48–0.62 thresholds in Appendix B instead abstain on diagnosis-margin confidence and never alter the 0.4 class-probability trigger.

Typed repair and one-object admission. At most 40 candidates arise from five operators: gap-bridge, strengthen-premise, specialize, split-branch, and inductive-step. Each candidate is Lean 4 type-checked with a 30-second timeout. Split-branch represents its two subclaims as one conjunction theorem; projections are local proof terms, and only the conjunction can enter the library. The operator therefore respects $K = 1$.

Adversarial novelty test. Type-correct candidates pass alpha-normalized syntactic deduplication, semantic-embedding comparison at $\delta_{\text{sem}} = 0.92$, and five generated paraphrase probes evaluated under the predeclared relaxed boundaries. A candidate is removed when any probe matches an accessible base- or induced-library statement, making the test adversarial rather than average-case. Across the three-run type-checked pool, 17.6% survive the offline audit, hence 82.4% are rejected. The selected online trajectory in Appendix Table 11 rejects $3295/3919 = 84.1\%$; it records a different, trajectory-specific pool.

Temporal future-reference ranking. A three-layer MLP assigns survivors score $R_\omega(\ell, \tau)$ from signature complexity, proof-gap context, neighboring library signatures, and operator type. The pairwise objective is

$$\mathcal{L}_{\text{rank}} = \sum_{(\ell^+, \ell^-)} \max\big(0, \epsilon - R_\omega(\ell^+, \tau) + R_\omega(\ell^-, \tau)\big), \tag{3}$$

where ℓ^+ is referenced by a later successful proof and ℓ^- completes its 12-round exposure without a reference. Temporal training replays only training families in chronological order; an update at training round r may use cohorts completed by $r-1$ and no later cohort. Development families fix the no-admit threshold before evaluation. The random-ranking control retains the full score only for this frozen admit/abstain decision and, on admission, samples uniformly from the same survivors. For the "no delayed supervision" intervention, the immediate label is one exactly when a checked candidate is explicitly referenced by the completed originating repair and zero for every other checked survivor; its MLP replaces R_ω under the same numeric threshold, survivor set, and cap. The ranker, diagnosis model, novelty rule, and threshold are frozen before either test benchmark is run, so test-family outcomes never enter an update.

Algorithm 1. BRIDGEBANK constrained delayed-credit gate

1. Fix the base library $\mathcal{L}_0$ and initialize the induced library $\mathcal{L}_{\mathrm{ind}} \leftarrow \emptyset$.
2. For each round $r = 1, \ldots, R$ and each goal $g \in \mathcal{G}$, run $\pi \leftarrow P_\theta(g \mid \mathcal{L}_0 \cup \mathcal{L}_{\mathrm{ind}})$.
3. If π fails with trace τ, compute $(\hat{y}, \hat{p}) \leftarrow D_\phi(\tau)$.
4. Only when $\hat{y}$ is missing auxiliary lemma and $\hat{p}_{y_1} > \gamma$, apply the typed repair operators to obtain candidate set C.
5. Type-check candidates: $C' \leftarrow \{\ell \in C : \textsc{Lean4Check}(\ell)\}$.
6. Novelty-filter survivors: $C'' \leftarrow \textsc{NoveltyFilter}(C', \mathcal{L}_0 \cup \mathcal{L}_{\mathrm{ind}})$.
7. If $C'' = \emptyset$, take the empty action; otherwise let $\ell^\star = \arg\max_{\ell \in C''} R_\omega(\ell, \tau)$ and take the empty action when its score is below the fixed no-admit threshold.
8. Otherwise update $\mathcal{L}_{\mathrm{ind}} \leftarrow \mathcal{L}_{\mathrm{ind}} \cup \{\ell^\star\}$ with the top survivor, admitting one persistent object.
9. During training replay only, update R_ω after a round from training-family cohorts that completed all 12 exposure rounds; freeze R_ω for development selection and test evaluation.

5 EXPERIMENTS

5.1 SETUP

Benchmarks and holdout. miniF2F contributes 488 competition-level Lean 4 test goals (Zheng et al., 2022), grouped into 96 held-out structural families. Mathlib-Structured contains 1,247 test theorems in 214 families selected for recurring signatures, with disjoint training and development partitions of 3012/418 and 612/86 goals/families. Synthetic-Gap removes known intermediates from 300 dependency-complete proofs and uses 180/72, 60/24, 60/24 train/development/test goals/families.

Family construction precedes every outcome. Goals are connected by declaration dependencies, shared alpha-normalized theorem schemas, or normalized goal signatures, and connected components are assigned intact to one split. The recurrence statistic combines rank-normalized train-corpus signature-neighbor density with rank-normalized dependency-pattern repetition; it is computed before test outcomes and partitions the Mathlib test set into fixed groups of $416/416/415$ goals. miniF2F is test-only for all learned components; model fitting uses training families, and thresholds and prompts use development families.

Systems and budgets. The common base is a Llama-3.1-70B proposal model (Grattafiori et al., 2024) interacting with the same Lean 4 tactic-state driver, premise index, initial library, and stopping implementation. It decodes at $T = 0.7$, top-$p = 0.95$. B1 performs prompt-only whole-proof repair and does not persist a helper. B2 applies up to 40 typed mutations and retains a verified helper used by the completed repair; B3 mines and retains helpers from solved traces. B4 expands the common premise-retrieval width by $4\times$; B5 substitutes ATP/hammer search with 120 seconds per goal; and B6 substitutes DeepSeek-Prover-V1.5 with 512 MCTS expansions (Xin et al., 2025); B4–B6 do not add persistent lemmas. BroadGrow-PF (B7) retains the common base, prompts, 40-candidate proposal cap, goal order, family splits, novelty test, and exposure horizon. It triggers on every policy-induced failure and retains the first verified novelty survivors in fixed generation order up to 8, with no reuse ranker; retained lemmas enter the same premise index before the next goal. BRIDGEBANK changes the trigger and retention rule and keeps at most one. Every system receives 576 end-to-end A100-hours, including generation, search, retries, ranking, checking, index updates, and exposure follow-up, on eight NVIDIA A100 80GB GPUs with CUDA 12.4.

Chronology, denominators, and uncertainty. For paired seeds 17/31/53, each policy begins from the same initial library and traverses the same fixed goal order for 12 induction rounds. Unsolved goals remain eligible in later rounds; solved goals are not counted again in solve but remain scheduled in exposure-only replay so an admitted lemma faces a common opportunity stream. A lemma retained in round r mutates the policy's library before the next goal and opens a cohort observed over exactly rounds $r + 1$ through $r + 12$. After the twelfth induction round, mutation stops and trailing

exposure-only rounds finish late cohorts. These rounds permit retrieval and proof measurement but no new admission or parameter update.

All test-time parameters are frozen. Solve uses distinct test goals; Reuse@12 and direct A/R/U use fully exposed admission cohorts; failed traces and Unused/100 are policy-specific. Accordingly, the broad-growth comparison reports direct counts rather than applying one policy's failure denominator to another. Means and standard deviations are seed-level. Paired differences use 10,000 bootstrap replicates with theorem family and seed as nested blocks; intervals are two-sided 95% percentile intervals. Win/loss/tie counts compare family-level outcomes after averaging within seed.

5.2 RQ1: Does Reuse-Guided Admission Improve the Resulting Library?

Table 2: Family-held-out results under the 576-A100-hour protocol. Solve, Reuse@12, and Short.$^+$ are seed-level percentages; Short.$^+$ is conditional on successful reruns that become shorter. B2 retains a verified mutation used by its repair, B3 retains a mined helper, and B7 and BridgeBank contrast broad and selective persistence in the shared stack; dashes denote policies that add no persistent lemma.

	miniF2F			Mathlib-Structured		
Method	Solve ↑	Reuse ↑	Short.$^+$	Solve ↑	Reuse ↑	Short.$^+$
Prompt repair (B1)	39.1±0.9	–	–	28.4±1.1	–	–
Mutation search (B2)	39.8±1.0	3.2	4.1	29.1±1.2	4.1	3.8
Heuristic mining (B3)	40.2±0.8	5.8	6.3	29.7±0.9	7.2	5.9
Retrieval $4\times$ (B4)	40.6±0.7	–	–	30.2±0.8	–	–
ATP/hammer (B5)	41.1±0.8	–	–	30.8±0.9	–	–
DeepSeek-V1.5 (B6)	41.4±0.6	–	–	31.2±0.7	–	–
BroadGrow-PF (B7)	41.8±0.8	7.6	6.9	31.9±0.9	8.7	8.1
BridgeBank	42.3±0.7	14.7	11.8	32.6±0.8	16.2	13.4

Table 3: Direct admission accounting over fully exposed cohorts. A/R/U are accepted/reused/unused lemmas pooled across three seeds; Cand./goal counts raw generated candidates. Direct counts avoid imposing a shared failure denominator on policy-dependent trajectories.

Benchmark	Policy	Accepted	Reused	Unused ↓	Cand./goal ↓
miniF2F	B7	1984	151	1833	29.8
miniF2F	BridgeBank	610	90	520	7.1
Mathlib-Structured	B7	3820	332	3488	31.6
Mathlib-Structured	BridgeBank	1006	163	843	8.4

Table 2 shows similar solve and an approximately twofold Reuse@12 increase over broad growth. Table 3 gives the operational consequence without mixing policy-specific failure denominators: selective persistence leaves $3.5\times$ fewer unused miniF2F lemmas and $4.1\times$ fewer on Mathlib-Structured. B7 obtains more reused lemmas in absolute count by exposing far more objects; BridgeBank instead increases the fraction that receive an observed future reference while producing a substantially smaller induced library.

The paired analysis separates current-goal success from the composition and size of persistent state. Against B7, BridgeBank's solve difference is $+0.5$ $[-0.1, 1.1]$ on miniF2F and $+0.7$ $[0.0, 1.4]$ on Mathlib-Structured; the reuse intervals separate at $+7.1$ $[5.5, 8.7]$ and $+7.5$ $[5.9, 9.1]$. The unused-count ratios of 0.28 $[0.24, 0.33]$ and 0.24 $[0.20, 0.29]$ show that the same compute envelope can produce markedly different persistent libraries even when solve is close.

The budget sweep preserves the separation: from $0.5\times$ to $1.5\times$ compute, solve deltas are $+0.1/+0.5/+0.4$ and reuse deltas are $+4.9/+7.1/+7.4$. Appendix E reports each component of the 576-hour ledger.

Table 4: Family-blocked paired differences (95% intervals) and family win/loss/tie counts on 96 miniF2F and 214 Mathlib-Structured families. Reuse differences are percentage points; the final row is the ratio of direct fully exposed unused counts. A win favors BRIDGEBANK: higher solve/reuse or a lower unused-count ratio.

Contrast	miniF2F estimate	W/L/T	Mathlib estimate	W/L/T
BRIDGEBANK−B6 solve	$+0.9$ $[0.2, 1.6]$	43/31/22	$+1.4$ $[0.6, 2.2]$	103/68/43
BRIDGEBANK−B7 solve	$+0.5$ $[-0.1, 1.1]$	38/34/24	$+0.7$ $[0.0, 1.4]$	89/79/46
BRIDGEBANK−B7 reuse	$+7.1$ $[5.5, 8.7]$	69/16/11	$+7.5$ $[5.9, 9.1]$	165/26/23
Unused-count ratio	0.28 $[0.24, 0.33]$	84/8/4	0.24 $[0.20, 0.29]$	194/13/7

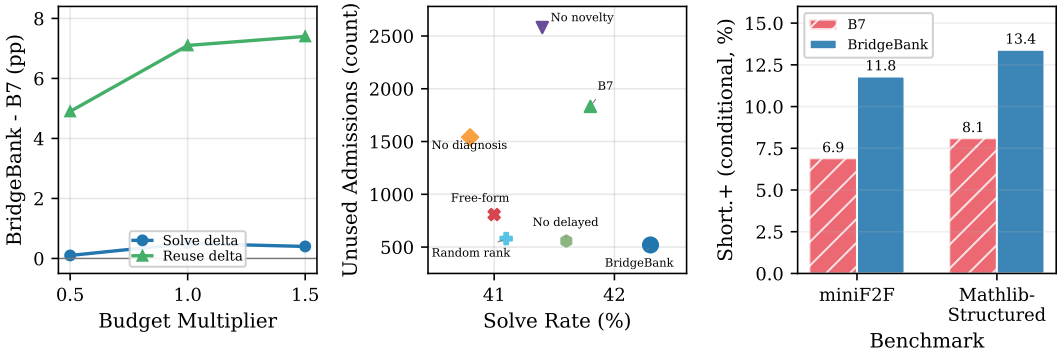


Figure 1: Delayed-utility evidence on miniF2F. Left: BRIDGEBANK−B7 solve and reuse differences at $0.5\times$ /$1\times$ /$1.5\times$ compute. Middle: solve versus direct fully exposed unused admissions; lower-right points are preferable, and the near-volume controls cluster around 557–580 unused objects. Right: descriptive Short.$^{+}$ on qualifying reused-proof reruns.

5.3 RQ2: How Do Gate Interventions Change Observed Reuse and Clutter?

Intervention semantics. Each row changes one named decision from the full gate. "No diagnosis" invokes the typed generator after every failed trace; "no novelty" passes every type-correct candidate to the same ranking and admission stage; "random reuse ranking" keeps the full gate's development-frozen null decision but samples uniformly whenever it admits; "free-form" replaces the five typed operators while preserving downstream filtering; and "no delayed supervision" scores the same survivors from immediate originating-repair references under the full gate's numeric threshold. The 610, 638, and 631 admissions for the full, random, and immediate-supervision rows are observed outcomes rather than volume-matched design constraints.

The ablation reveals two distinct patterns. First, solve places no novelty filtering (41.4%) above random ranking (41.1%), but observed reuse and policy-specific unused burden reverse the order: 6.2% and 63.4 versus 9.1% and 14.2 per 100 failures. This is the empirical ordering disagreement anticipated by Proposition 1: solve alone does not determine the persistence coordinates. Second, the full, random-rank, and immediate-supervision rows expose 7.1 candidates per goal and retain 610, 638, and 631 lemmas; their observed reuse is 14.7%, 9.1%, and 11.8%, with 520, 580, and 557 unused objects. Their similar realized admission counts make these rows descriptive evidence about ranking rather than retention volume.

5.4 RQ3: Where Is the BridgeBank–BroadGrow Contrast Largest?

In the low-recurrence stratum, solve changes by $+0.1$ $[-0.5, 0.7]$ and reuse by $+2.6$ $[1.0, 4.2]$. The corresponding mid-stratum values are $+0.6$ $[0.0, 1.2]$ and $+6.7$ $[4.9, 8.5]$; high recurrence reaches $+1.4$ $[0.7, 2.1]$ and $+11.5$ $[9.1, 13.9]$. The largest observed BRIDGEBANK−B7 reuse difference therefore occurs in the predeclared high-recurrence stratum.

Table 5: miniF2F ablations over three seeds. Reuse is Reuse@12; Cand. counts raw candidates per goal; Audit is the offline novelty-pass percentage. Every A/R/U cohort is fully exposed for 12 rounds. Intervals are family-blocked marginal 95% intervals, not paired differences between rows.

Ablation	Solve	Reuse [95%]	Cand.	Audit
Full BRIDGEBANK	42.3	14.7 [13.1,16.3]	7.1	17.6
No diagnosis gate	40.8	8.3 [7.1,9.6]	24.6	8.1
No novelty filter	41.4	6.2 [5.1,7.3]	40.0	4.3
Random reuse ranking	41.1	9.1 [7.8,10.5]	7.1	17.6
Free-form generation	41.0	10.4 [9.0,11.9]	12.8	11.2
No delayed supervision	41.6	11.8 [10.2,13.4]	7.1	17.6

Ablation	Failed traces	A/R/U	Unused/100 [95%]
Full BRIDGEBANK	4059	610/90/520	12.8 [11.9,13.8]
No diagnosis gate	4084	1680/139/1541	37.7 [35.9,39.6]
No novelty filter	4068	2751/171/2580	63.4 [61.0,65.8]
Random reuse ranking	4076	638/58/580	14.2 [13.2,15.3]
Free-form generation	4070	902/94/808	19.9 [18.5,21.4]
No delayed supervision	4065	631/74/557	13.7 [12.7,14.8]

6 PERSISTENCE CONTROL AS A SYSTEM PRIMITIVE

The recurrence-conditioned ordering supplies an implementable routing signal. A system can estimate structural recurrence before mutation and condition its required predicted-reference score on that estimate. This control layer composes with any checked repair operator: a checker establishes eligibility, while completed downstream exposure supplies the training target for retention.

The experiments answer the three research questions with complementary evidence. B7 measures the end-to-end consequence of broad versus selective growth, direct A/R/U counts expose the library produced, near-volume interventions describe ranker behavior, and recurrence strata locate effect heterogeneity. Appendix C additionally instantiates the same top-one-or-none interface in Isabelle/HOL.

Scope of the evidence. The matched B7 analysis establishes the reuse and library-size separation for family-held-out formal workloads, three paired seeds, and a fully observed 12-round horizon; the near-volume and Isabelle results are descriptive at their reported resolution. Reuse@12 records references on each policy's induced trajectory, and Short.$^{+}$ summarizes only successful masked

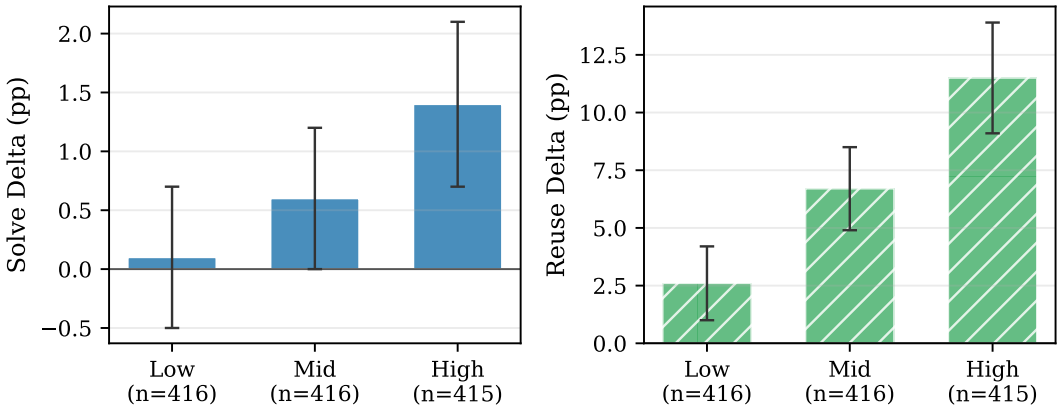


Figure 2: Mathlib-Structured BRIDGEBANK−B7 differences by predeclared recurrence tertile ($n = 416/416/415$ goals). Recurrence combines rank-normalized train-corpus signature neighbors and repeated dependency patterns before test outcomes. Whiskers show family-blocked 95% intervals.

reruns that become shorter, so we keep both quantities distinct from a causal no-admission effect. The completed-cohort protocol directly supports comparisons at the reported horizon and capacity.

7 CONCLUSION

BRIDGEBANK inserts a learned future-reference decision between verification and shared retrieval state. Across two family-held-out Lean evaluations, it raises observed Reuse@12 by 7.1–7.5 points over broad growth, leaves 3.5–$4.1\times$ fewer fully exposed admissions unused, and changes solve by $+0.5$ and $+0.7$ points. At similar realized retention counts, its 14.7% reuse point estimate exceeds random ranking's 9.1% and immediate supervision's 11.8%; the largest paired policy difference occurs in the high-recurrence stratum. A growing prover should earn persistence with evidence of future use, not verification alone.

AI USE STATEMENT

Generative AI tools assisted in drafting, editing, and presentation of this manuscript, as well as its submission-format preparation. The authors reviewed all AI-assisted work and take responsibility for the paper's technical content and final text.

REFERENCES

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen Marcus McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4WnqRR915j.

Anne Baanen, Matthew Robert Ballard, Johan Commelin, Bryan Gin-ge Chen, Michael Rothgang, and Damiano Testa. Growing Mathlib: Maintenance of a large scale mathematical library. In *Intelligent Computer Mathematics*, volume 16136 of *Lecture Notes in Computer Science*, pp. 51–70. Springer, 2026. doi: 10.1007/978-3-032-07021-0_4. URL https://doi.org/10.1007/978-3-032-07021-0_4.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning*, pp. 41–48, 2009. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

Edmund Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veith. Counterexample-guided abstraction refinement for symbolic model checking. *Journal of the ACM*, 50(5):752–794, 2003. doi: 10.1145/876638.876643. URL https://doi.org/10.1145/876638.876643.

Emily First, Markus N Rabe, Talia Ringer, and Yuriy Brun. Baldur: Whole-proof generation and repair with large language models. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 1229–1241, 2023. doi: 10.1145/3611643.3616243. URL https://doi.org/10.1145/3611643.3616243.

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://doi.org/10.1007/s10817-020-09580-x.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. doi: 10.48550/arXiv.2407.21783. URL https://arxiv.org/abs/2407.21783.

Jesse Michael Han, Jason Rute, Yuhuai Wu, Edward Ayers, and Stanislas Polu. Proof artifact co-training for theorem proving with language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=rpxJc9j04U.

Thomas Hubert, Rishi Mehta, Laurent Sartran, Miklós Z. Horváth, Goran Žužić, Eric Wieser, Aja Huang, Julian Schrittwieser, Yannick Schroecker, Hussain Masoom, Ottavia Bertolli, Tom Zahavy, Amol Mandhane, Jessica Yung, Iuliya Beloshapka, Borja Ibarz, Vivek Veeriah, Lei Yu, Oliver Nash, Paul Lezeau, Salvatore Mercuri, Calle Sönne, Bhavik Mehta, Alex Davies, Daniel Zheng, Fabian Pedregosa, Yin Li, Ingrid von Glehn, Mark Rowland, Samuel Albanie, Ameya Velingker, Simon Schmitt, Edward Lockhart, Edward Hughes, Henryk Michalewski, Nicolas Sonnerat, Demis Hassabis, Pushmeet Kohli, and David Silver. Olympiad-level formal mathematical reasoning with reinforcement learning. *Nature*, 651(8106):607–613, 2026. doi: 10.1038/s41586-025-09833-y. URL https://doi.org/10.1038/s41586-025-09833-y.

Geoffrey Irving, Christian Szegedy, Alexander A Alemi, Niklas Een, Francois Chollet, and Josef Urban. DeepMath – deep sequence models for premise selection. In *Advances in Neural Information Processing Systems*, volume 29, pp. 2235–2243, 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/hash/f197002b9a0853eca5e046d9ca4663d5-Abstract.html.

Albert Q Jiang, Wenda Li, Szymon Tworkowski, Konrad Czechowski, Tomasz Odrzygóźdź, Piotr Miłoś, Yuhuai Wu, and Mateja Jamnik. Thor: Wielding hammers to integrate language models and automated theorem provers. *Advances in Neural Information Processing Systems*, 35:8360–8373, 2022. doi: 10.52202/068431-0608. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/377c25312668e48f2e531e2f2c422483-Abstract-Conference.html.

Albert Q Jiang, Sean Welleck, Jin Peng Zhou, Wenda Li, Jiacheng Liu, Mateja Jamnik, Timothée Lacroix, Yuhuai Wu, and Guillaume Lample. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=SMa9EAovKMC.

Kazumi Kasaura, Naoto Onda, Yuta Oriike, Masaya Taniguchi, Akiyoshi Sannai, and Sho Sonoda. Discovering new theorems via LLMs with in-context proof learning in Lean. *arXiv preprint arXiv:2509.14274*, 2025. doi: 10.48550/arXiv.2509.14274. URL https://arxiv.org/abs/2509.14274.

Jongyoon Kim, Hojae Han, and Seung-won Hwang. Benchmarking testing in automated theorem proving. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 2241–2260, San Diego, California, USA, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-industry.150. URL https://aclanthology.org/2026.acl-industry.150/.

Adarsh Kumarappan, Mohit Tiwari, Peiyang Song, Robert Joseph George, Chaowei Xiao, and Anima Anandkumar. Leanagent: Lifelong learning for formal theorem proving. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/b67c77f8db8b991d73d6f2e16f491840-Abstract-Conference.html.

Guillaume Lample, Timothee Lacroix, Marie-Anne Lachaux, Aurelien 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, pp. 26337–26349, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/a8901c5e85fb8e1823bbf0f755053672-Abstract.html.

Claire Le Goues, Michael Pradel, and Abhik Roychoudhury. Automated program repair. *Communications of the ACM*, 62(12):56–65, 2019. doi: 10.1145/3318162. URL https://doi.org/10.1145/3318162.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision*, pp. 2980–2988, 2017. doi: 10.1109/ICCV.2017.324. URL https://doi.org/10.1109/ICCV.2017.324.

Xinyu Liu, Zixuan Xie, Amir Moeini, Claire Chen, Shuze Daniel Liu, Yu Meng, Aidong Zhang, and Shangtong Zhang. MathlibLemma: Folklore lemma generation and benchmark for formal mathematics. *arXiv preprint arXiv:2602.02561*, 2026. doi: 10.48550/arXiv.2602.02561. URL https://arxiv.org/abs/2602.02561.

Sarah Loos, Geoffrey Irving, Christian Szegedy, and Cezary Kaliszyk. Deep network guided proof search. In *LPAR-21. 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, volume 46 of *EPiC Series in Computing*, pp. 85–105, 2017. doi: 10.29007/8mwc. URL https://easychair.org/publications/paper/ND13.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Stanislas Polu and Ilya Sutskever. Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*, 2020. doi: 10.48550/arXiv.2009.03393. URL https://arxiv.org/abs/2009.03393.

Stanislas Polu, Jesse Michael Han, Kunhao Zheng, Mantas Baksys, Igor Babuschkin, and Ilya Sutskever. Formal mathematics statement curriculum learning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=-P7G-8dmSh4.

Haiming Wang, Huajian Xin, Chuanyang Zheng, Zhengying Liu, Qingxing Cao, Yinya Huang, Jing Xiong, Han Shi, Enze Xie, Jian Yin, Zhenguo Li, and Xiaodan Liang. LEGO-Prover: Neural theorem proving with growing libraries. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/85dca46374dc0f27b4bb5f265b3d17f0-Abstract-Conference.html.

Minchao Wu, Michael Norrish, Christian Walder, and Amir Dezfouli. Tacticzero: Learning to prove theorems from scratch with deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 9330–9342, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/4dea382d82666332fb564f2e711cbc71-Abstract.html.

Yuhuai Wu, Albert Q Jiang, Wenda Li, Markus N Rabe, Charles Staats, Mateja Jamnik, and Christian Szegedy. Autoformalization with large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 32353–32368, 2022. doi: 10.52202/068431-2344. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/d0c6bc641a56bebee9d985b937307367-Abstract-Conference.html.

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, Aidan M. Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan J. Prenger, and Anima Anandkumar. Leandojo: Theorem proving with retrieval-augmented language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 21573–21612, 2023. doi: 10.52202/075280-0944. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/4441469427094f8873d0fecb0c4e1cee-Abstract-Datasets_and_Benchmarks.html.

Youyuan Zhang, Jialiang Sun, Hangrui Bi, Chuqin Geng, Wenjie Ma, Zhaoyu Li, and Xujie Si. DreamProver: Evolving transferable lemma libraries via a wake-sleep theorem-proving agent. *arXiv preprint arXiv:2604.26311*, 2026. doi: 10.48550/arXiv.2604.26311. URL https://arxiv.org/abs/2604.26311.

Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. minif2f: A cross-system benchmark for formal Olympiad-level mathematics. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=9ZPegFuFTFv.

A Extended Experimental Details

The diagnosis model is a 6-layer transformer encoder with hidden size 512, optimized by AdamW at 2×10^{-4} (Loshchilov & Hutter, 2019). The reuse predictor is a 3-layer MLP with margin $\epsilon = 0.1$. Training proceeds in curriculum order from Synthetic-Gap to real traces (Bengio et al., 2009). Real-trace labels come from two annotators with initial agreement $\kappa = 0.79$ and adjudication by discussion. The temporal predictor is then trained on completed training-family cohorts; the immediate-supervision control instead uses originating-repair support. Development families select the trigger, prompt, semantic, and no-admit thresholds. Table 6 records the family boundary used by every policy.

Table 6: Family-disjoint data use. A dash indicates that miniF2F is reserved entirely for test. Each cell is goals/families; the Synthetic-Gap row sums to its 300-problem benchmark definition.

Corpus	Train	Development	Test
miniF2F	–	–	488/96
Mathlib-Structured	3012/418	612/86	1247/214
Synthetic-Gap	180/72	60/24	60/24

For each seed, policies receive identical theorem order and initial library. Training checkpoints and thresholds are frozen before test; no scorer update occurs during either test stream. Admissions mutate only their policy's library, and late cohorts receive mutation-free exposure rounds until all 12 observations are complete. Bootstrap analysis is performed after all runs. B2 persists a verified mutation used in its successful repair, and B3 persists a helper mined from a completed proof. The B7 adaptation shares generation prompts, type-check limits, novelty rules, and index updates with BridgeBank, triggers on every policy-induced failure, and retains the first verified novelty survivors in fixed generation order up to its per-failure cap.

B Gate Diagnostics

Table 7: Diagnosis on 501 real Lean 4 failed traces. ECE is expected calibration error.

Class	n	Precision	Recall	F1	ECE
Missing auxiliary lemma	142	0.74	0.78	0.76	0.051
Retrieval failure	98	0.71	0.68	0.69	0.063
Underpowered search	127	0.68	0.72	0.70	0.054
Malformed decomposition	83	0.65	0.62	0.63	0.072
False target	51	0.78	0.82	0.80	0.041
Macro average	501	0.71	0.72	0.71	0.058

Macro-F1 0.71 and ECE 0.058 support selective routing while exposing the classes where abstention matters. Table 8 replaces scalarized local utility with the direct decision quantities used in the paper.

Table 8: Diagnostic margin-threshold sweep on the confounded diagnosis set ($n = 287$). P/R is missing-bridge precision/recall; A/R/U reports admissions/reused/unused admissions per 100 decisions. These τ settings do not alter the main policy's $\gamma = 0.4$ trigger.

Policy	Solve	P/R	Misdirected/100	A/R/U per 100
No abstention ($\tau = 0$)	27.2	0.49/1.00	38.6	48.1/4.0/44.1
Abstain at $\tau = 0.48$	27.9	0.61/0.87	26.4	34.7/4.4/30.3
Abstain at $\tau = 0.55$	28.1	0.69/0.78	21.3	27.5/4.6/22.9
Abstain at $\tau = 0.62$	27.6	0.77/0.61	16.8	19.2/3.7/15.5

This margin-threshold sweep is diagnostic only and does not alter BridgeBank's $\gamma = 0.4$ test-time trigger. Within the diagnostic slice, $\tau = 0.55$ reduces both misdirected operations and unused

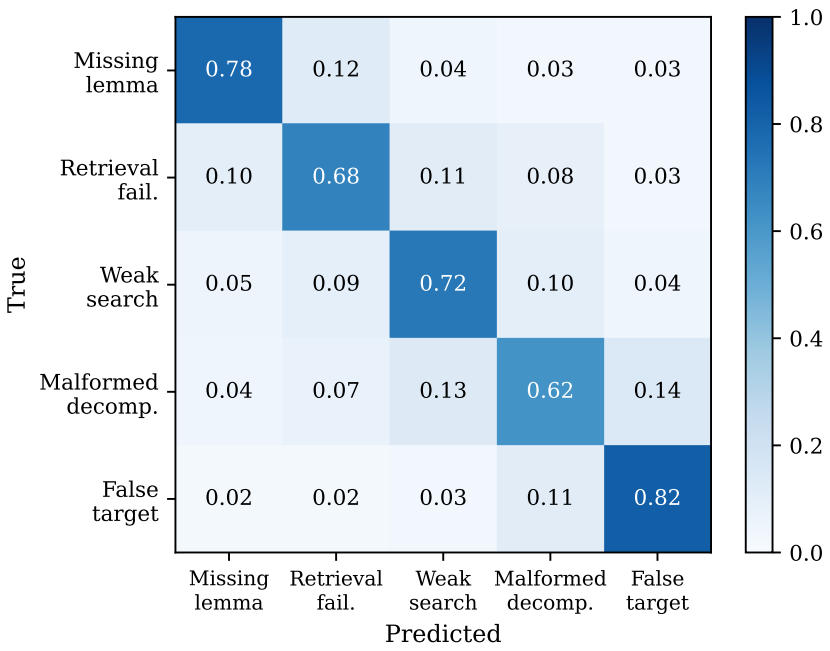


Figure 3: Row-normalized diagnosis confusion on 501 failed traces. The largest off-diagonal cell is malformed decomposition predicted as false target (14.0% of that row).

admissions while improving solve; the stricter setting further reduces misdirection at lower missing-bridge recall.

Table 9: Novelty-filter errors on 500 lemma pairs. Controlled uses oracle labels; paraphrase-heavy adds rephrasings and retrieval leakage. Rates are percentages.

Audit set	Pairs	False merge	False split
Controlled	500	2.1	4.7
Paraphrase-heavy	500	3.8	7.2

The paraphrase-heavy audit makes the novelty ablation interpretable: semantic leakage raises both error modes, while the no-filter intervention has little solve difference and a fivefold larger direct unused burden.

C Descriptive Isabelle/HOL Instantiation

Table 10: Descriptive AFP-obligation instantiation in Isabelle/HOL with Sledgehammer under the 576-A100-hour cap. Values are percentages.

Method	Solve	Reuse
Prompt repair (B1)	31.4±1.2	–
Mutation search (B2)	32.1±1.0	2.9
Heuristic mining (B3)	32.6±0.9	5.4
Retrieval 4× (B4)	32.9±1.1	–
ATP/Sledgehammer-heavy (B5)	33.2±0.8	–
BridgeBank	33.5±0.9	9.3

This operating point reaches 9.3% observed reuse and 33.5% solve. The percentages document a second-assistant instantiation of the top-one-or-none interface rather than a cross-assistant effect comparison.

D Additional Result Breakdowns

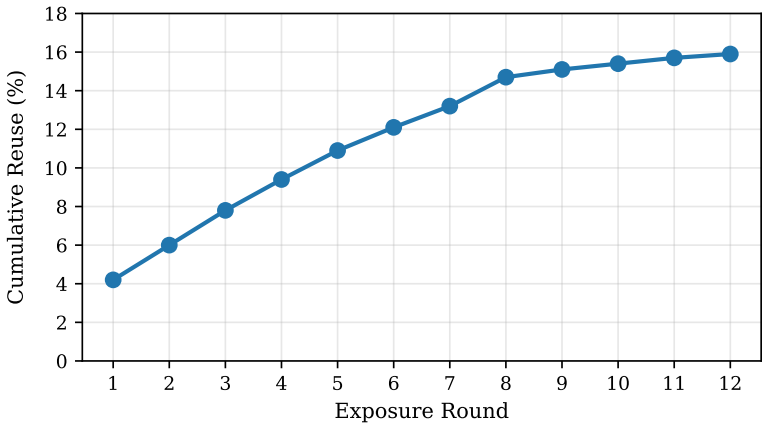


Figure 4: One retained miniF2F BridgeBank trajectory reaches 15.9% cumulative reuse after 12 exposure rounds. This trajectory-level view is distinct from the 14.7% three-seed Reuse@12 mean in Table 2.

Table 11: Round-level BridgeBank induction audit for the selected miniF2F seed in Figure 4. Failures are policy-specific diagnosed traces, throughput is accepted per such trace, and cumulative reuse is an illustrative trajectory statistic rather than the pooled main-table denominator.

Round	Failures	Raw cand.	Type-OK	Novel	Accepted	Throughput	Cum. reuse
1	142	891	412	78	12	0.085	4.2
2	138	867	398	71	16	0.116	6.0
3	131	824	381	64	19	0.145	7.8
4	127	798	369	61	21	0.165	9.4
5	119	748	346	57	23	0.193	10.9
6	112	704	325	52	21	0.188	12.1
7	108	679	314	48	19	0.176	13.2
8	103	648	299	44	17	0.165	14.7
9	99	621	288	41	16	0.162	15.1
10	95	596	274	38	15	0.158	15.4
11	91	571	262	36	14	0.154	15.7
12	88	548	251	34	13	0.148	15.9

The table contains 3,919 type-checked candidates and 624 novelty survivors, an online rejection rate of 84.1%. Acceptance remains bounded as diagnosed failures decline, and the selected trajectory's reuse saturates from exposure rounds 10–12. Its induction counts are neither multiplied into the pooled A/R/U table nor used to normalize B7; the figure supplies a temporal mechanism view alongside the three-seed mean.

Group/ring theory and combinatorics have the largest reported solve and reuse differences, while analysis/topology has the smallest. The count-weighted solve average rounds to the 32.6% main-table result.

Table 12: Descriptive Mathlib-Structured domain breakdown. Solve cells count-weight to the 32.6% main result. Short.$^+$ is conditional positive proof shortening; the second panel reports BRIDGEBANK−B7 differences in solve and reuse.

Domain	n	Solve	Reuse	Short.$^+$	Diag. F1	Novel
Group/Ring theory	287	35.6	19.8	15.2	0.74	21.3
Combinatorics	241	34.2	16.4	13.8	0.72	18.7
Number theory	198	32.3	14.1	11.2	0.70	16.4
Linear algebra	264	31.6	12.8	10.1	0.69	14.9
Analysis/Topology	257	29.1	7.3	5.4	0.66	11.2

Domain	Δ solve	Δ reuse
Group/Ring theory	+1.2	+10.8
Combinatorics	+0.9	+8.6
Number theory	+0.7	+7.1
Linear algebra	+0.5	+5.7
Analysis/Topology	+0.2	+2.4

Table 13: Compute accounting at the $1\times$ cap. Breakdown order is diagnosis/generation/novelty-and-ranking/prover-and-search/checking, in A100-hours. Operational counts are wall-clock hours/model calls/solver calls/Lean checks/candidates.

Policy	Component breakdown	Total	Operational counts
BRIDGEBANK	38/171/52/267/48	576	74.6/18340/52810/125760/14118
BroadGrow-PF	25/205/18/279/49	576	73.9/46780/51940/211340/60214

Table 14: miniF2F BRIDGEBANK−B7 budget robustness. Entries are ordered $0.5\times/1\times/1.5\times$ (288/576/864 A100-hours); both rows are percentage-point differences.

Metric	$0.5\times/1\times/1.5\times$
Solve	+0.1/+0.5/+0.4
Reuse@12	+4.9/+7.1/+7.4

E COMPUTE ROBUSTNESS

The reuse separation survives the budget sweep and accounting decomposition. BroadGrow-PF's diagnosis and novelty-and-ranking entries are accounting buckets for trace processing and shared novelty/index updates; it still generates after every policy-induced failure and has no learned reuse ranker. BRIDGEBANK spends more on admission control and fewer calls on generation, converting comparable total compute into a smaller persistent library with higher reuse.

F QUALITATIVE FAILURE ANALYSIS

In a descriptive sample of 60 accepted lemmas, collaborators label 18 trivial, 25 proof-local, 12 domain-reusable, and 5 abstraction-level. The resulting 43 local versus 17 reusable or abstraction-level assignments reinforce the need for direct use-based accounting after verification.

G PROOF OF THE SOLVE–PERSISTENCE RANK-INVERSION PROPOSITION

Proof. Subtracting Eq. 1 gives $\mathcal{U}_H(b)-\mathcal{U}_H(a) = D_b - D_a - \alpha(\Delta S_a - \Delta S_b)$. The stated conditions make this difference positive while solve ranks a above b; rearrangement proves the converse. □