ABRL: A SOURCE-ALIGNED LEAN 4 PROOF ATLAS FOR BANDIT AND REINFORCEMENT LEARNING THEORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding a new bandit or reinforcement-learning result requires identifying which arguments are inherited, which assumptions change, and which proof obligations remain unmatched. We present BanditRLlib, a source-aligned Lean 4 library organized as a proof atlas for this comparison, within the ABRL project. The library formalizes a classical bandit backbone, five end-to-end setting extensions, and a known-reward Hoeffding UCB-VI development, pairs each source-facing theorem with a plain-language proof and its exact Lean declaration, and exposes two complementary views: a compiler-derived dependency graph showing what is formally reused, and a map of conceptual correspondences across settings recorded separately from checked reuse. Two results illustrate the atlas. Heavy-tailed and causal bandits reuse the same two concentration lemmas, called by name, while sharing neither policy nor criterion. A countable-node kernel adapter for hierarchical optimistic optimization removes an extraneous measurability premise introduced by our initial formalization, without changing the source's scope, and a finite certificate shows that a printed pull-count coefficient in the heavy-tailed source fails, with an amended bound proved for the unchanged policy. Source correspondence is reviewed and recorded for every source-facing statement through source-blind reconstruction and explicit discrepancy records. The result is a checked reference frame for reading new work against classical theory: what is inherited, what is recombined, and what remains to be proved. The library, its records, and its tools are available at https://anonymous.4open.science/r/peer-review-artifact-2027.

1 INTRODUCTION

Bandit theory grows by variation and recombination. A concentration argument that first appears for a finite stochastic bandit reappears under heavy-tailed rewards, causal interventions, delayed feedback, combinatorial actions, or a reinforcement-learning (RL) value function, each time with just enough of the model changed that the connection is easy to miss (Lattimore & Szepesvári, 2020; Slivkins, 2019; Sutton & Barto, 2018). Many new results recombine familiar proof blocks under a modified feedback model, geometry, robustness requirement, constraint, or objective, and a few introduce a genuinely new bridge. For a reader, the hard part is seeing which is which: which assumptions changed, which proof components were inherited, and which step actually required new mathematics.

That reconstruction is becoming a larger part of the research burden. AI-assisted systems now draft manuscripts and formalize research papers and textbook-scale mathematics (Zhang et al., 2026c; Gloeckle et al., 2026; Min et al., 2026), so the cost of producing technically plausible artifacts is falling while the human cost of checking and digesting them is not. Proof assistants such as Lean 4 with Mathlib (de Moura & Ullrich, 2021; The mathlib Community, 2020) certify that a formal derivation is valid. Checking a derivation is not the same as understanding it, and a certified derivation does not by itself say how a new proof fits into what a field already knows. Verification does, however, provide a substrate on which explanation, comparison, and reuse can be built.

ABRL builds that substrate for bandit and RL theory. Its central artifact, BanditRLlib, is a Lean 4 library organized around the field's textbook backbone. Each reviewed result is paired with a readable

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.

mathematical explanation, and the same formal declarations are exposed through a compiler-derived Lean Graph, a conceptual mechanism hypergraph of recurring proof mechanisms, and an assumption-indexed atlas, BanditRLwiki, with a Frontier of named open leaves. When a new bandit proof reuses a known concentration lemma but changes only the sampling model, that reuse should be visible rather than buried under new notation. When two settings share a mechanism but the transfer breaks at one hypothesis, that failure boundary should be visible. When a new bridge is required, the graph should show where it enters. A beginner can move from a textbook sentence to the Lean lemma behind it, an expert can compare assumptions and proof branches across settings, and a contributor can overlay a new theorem on the existing graph and see which leaves are missing. None of this assigns a novelty score, and a missing leaf is a gap relative to the current library, not evidence of mathematical novelty. It makes the anatomy of a proof inspectable, so that the evidence for expert judgment is on the page. For these comparisons to mean anything, the formal statements must remain faithful to their sources, and ABRL treats source alignment as a correctness condition on the growing library, enforced by a documented semantic round trip.

This paper makes three contributions.

1. **A source-aligned formal backbone.** We construct BanditRLlib, a Lean 4 library covering a classical bandit backbone, five end-to-end setting extensions, and a known-reward Hoeffding UCB-VI development, with explicit records of amendments, specializations, and incomplete routes (Sections 3 and 6).
2. **A proof atlas with explicit evidence levels.** We connect source-facing theorem pages to compiler-extracted dependencies, assumption-indexed setting records, and a separate map of conceptual correspondences, distinguishing checked reuse from candidate transfer and unmatched proof obligations (Sections 3 and 4).
3. **Machine-checked reuse and abstraction repair.** We expose the exact concentration lemmas reused by heavy-tailed and causal bandits and construct a countable-node kernel adapter that removes an extraneous global measurability premise from our initial HOO formalization. We additionally certify a finite-instance failure of the printed heavy-tailed pull-count coefficient and prove an amended bound for the unchanged policy (Sections 5 and 6.3).

2 Related work

Domain libraries and blueprints. Formal libraries for learning theory prove and organize reusable mathematics: Rademacher complexity and Dudley's entropy integral (Sonoda et al., 2025), Gaussian concentration, chaining, and localized least squares with a declaration-to-textbook table (Zhang et al., 2026b), and convergence of temporal-difference and Q-learning under Markovian sampling (Zhang, 2025). Blueprint projects connect natural-language and formal proofs for one theorem (Massot, 2020; Becker et al., 2024; Dillies et al., 2023; Scholze, 2022), and LeanArchitect manages and synchronizes blueprint data from Lean developments (Zhu et al., 2026). BanditRLlib follows this practice and adds a domain-level organization: adaptive interaction processes, regret criteria that differ by setting, and lower-bound machinery, indexed by setting and proof mechanism, with reuse across settings recorded at the level of declarations and results compared only under a common assumption and criterion signature.

Autoformalization at theory scale. Recent systems formalize whole textbooks and corpora (Gloeckle et al., 2026; Wang et al., 2026b; Feng et al., 2026; Wang et al., 2026a; Meadows et al., 2026), multi-agent harnesses formalize research papers with target review and CI-gated rounds (Zhang et al., 2026c; Milikic et al., 2026; Pegden, 2026), Min et al. (2026) argue for theory-level knowledge bases, and ProofFlow and Proof-Refactor evaluate and restructure generated proofs (Cabral et al., 2026; Fu et al., 2026). These systems optimize formalization generation and long-horizon execution. ABRL uses formalization to build a persistent, navigable mathematical substrate for one theory community, with explanations and graphs kept synchronized with the formal content (Appendix H).

Statement fidelity. The problem of a formal statement that does not say what its informal source says was framed by Adams (2016), and recent work measures the fidelity of generated statements with learned scores, equivalence checks, and perturbation tracking (Lu et al., 2025; Liu et al., 2025; 2026; Poiroux et al., 2025; Wu et al., 2026; Mohammad & Sheikh, 2026; Zhang et al., 2026a; Chen et al., 2026; Cornish et al., 2026; Gui et al., 2026). Compilation cannot replace a judgment of semantic

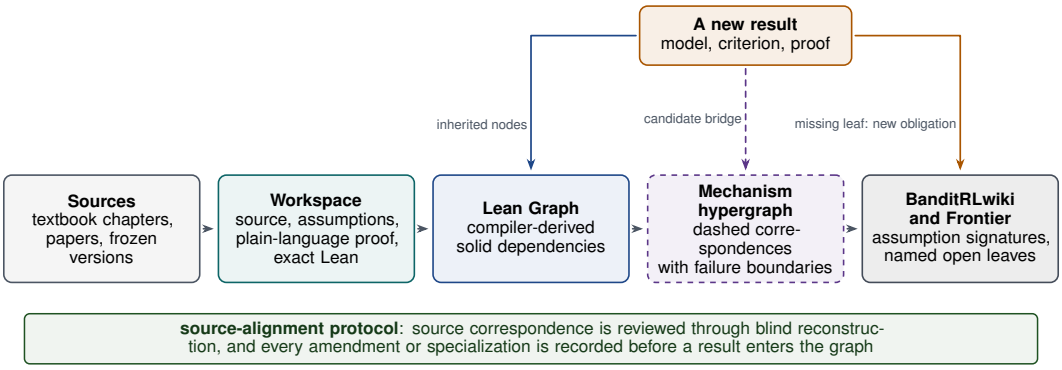

Figure 1: How BanditRLlib helps read a new result. Source theorems enter the workspace; the compiled declarations form the Lean Graph (solid, compiler-derived edges); recurring proof mechanisms are collected in the mechanism hypergraph (dashed correspondences, each with a failure boundary); settings, assumptions, and open leaves are indexed in BanditRLwiki and its Frontier. A new result overlaid on these views shows its inherited nodes, its candidate bridges, and the missing leaves that are its new obligations. The source-alignment protocol protects every stage.

correspondence, and the atlas is only useful while the natural-language theorem, the Lean theorem, the assumptions, and the source anchor stay aligned. We use these ideas as review tools inside the library.

3 BanditRLlib at a glance

BanditRLlib is built on Mathlib (The mathlib Community, 2020). Figure 1 shows how it is read. Source theorems enter a workspace that pairs the mathematics with exact Lean, the compiled declarations form the Lean Graph, recurring mechanisms across settings are collected in the Functor Hypergraph, and the settings themselves, with their assumptions and open problems, are indexed in BanditRLwiki and its Frontier. All of these views are browsable at `https://iclr-artifact2027.github.io/ABRL-ICLR/`.

Textbook backbone. The library's spine is the standard textbook (Lattimore & Szepesvári, 2020): teaching developments of explore-then-commit, UCB, OFUL, Thompson sampling, EXP3, and Tsallis-entropy FTRL, its lower-bound theory from the information-theoretic tools to the minimax, instance-dependent, and high-probability bounds, and a finite-horizon RL book whose developments run from Bellman optimality to a UCB-VI regret bound. Representative developments from the research literature, five bandit settings and three routes in progress, are attached to this spine.

Workspace. Every reviewed theorem has a page with five parts. **Source** gives the statement with edition and anchor. **Assumptions** lists the explicit differences between the source's hypotheses and the formal ones. **Proof** is a natural-language account with displayed formulas. **Lean** is the exact statement and proof. **Dependencies** names the library results the proof uses, together with a checked instantiation that applies the theorem to a concrete instance. Identifiers and review records are folded by default, so the first view is the mathematics.

Lean Graph. The dependency graph is extracted from the compiled library. Solid edges are compiler-derived, and the graph distinguishes a reference in a theorem's statement, which tells a reader what definitions the statement uses, from a reference only in its proof, which tells a reader which results the proof calls. It is disclosed progressively, from chapters and modules to individual declarations and open leaves, which compresses the presentation and not the proofs. This graph is the trusted backbone against which every conceptual correspondence is measured.

Mechanism hypergraph. Proof mechanisms recur across settings after the space, the feedback model, or the objective changes: a confidence set followed by optimism and a regret charge, an alternative environment followed by an information budget and a lower bound, a regularized potential followed by one-step stability, a robust estimator followed by a confidence object, metric localization, and

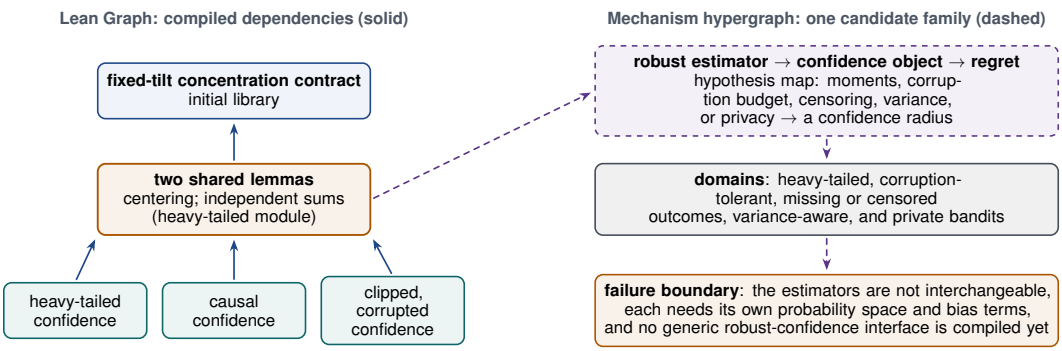

Figure 2: The same example in both graphs. Left: the Lean Graph records what is actually reused. Three confidence proofs call the same two lemmas, which instantiate the fixed-tilt contract, and every edge is a compiled dependency. Right: the mechanism hypergraph proposes the robust-estimator family that these proofs and three further settings may share, names the hypothesis map, and records the failure boundary. Its edges stay dashed until a compiled adapter turns the family into a solid edge.

others. The hypergraph records fifteen such candidate families. Each carries its domains, a hypothesis map and a conclusion map across those domains, the Lean declarations that could substantiate it, and a failure boundary stating where the analogy breaks. Its edges are dashed. They are proposed structural analogies, not theorem dependencies and not certified categorical functors, and a family becomes a solid edge only when a compiled adapter turns it into one. A technique map of twenty-five proof mechanisms links each setting to the families that apply to it. Figure 2 shows the two graphs on one example: the Lean Graph records that three proofs call the same two lemmas, and the hypergraph proposes the family those proofs and three further settings may share, with the boundary at which the analogy fails.

BanditRLwiki and Frontier. A new result is compared to the right classical case only if assumptions and objectives are normalized first. BanditRLwiki indexes thirteen cases in seven families, each with a twelve-field signature: the exact setting, assumptions, feedback structure, algorithm, criterion, expectation or high probability, parameter dependence, upper and lower bounds with their conditions, computational and oracle requirements, source and version, and the Lean declarations with their completion boundary. Each case records separately whether the literature is open and whether the formalization is open. The Frontier lists these open leaves by name, together with the source-audit queue and the partial routes, so that "in progress" always names a leaf.

How a new result enters. A contributor locates the case whose signature matches the new setting, formalizes the statement against the library, and overlays it on the graph. Nodes already present are the inherited mathematics. Hypergraph families that touch the setting are candidate bridges to other techniques. Leaves that no existing declaration covers are the model-specific arguments that remain to be proved. The protocol of Section 6.3 checks the statement before its node is trusted. As a worked case, overlay the clipped estimator of Section 5.4. Its inherited nodes are the fixed-tilt contract, the two concentration lemmas, and the heavy-tailed confidence route. Its candidate bridge is the robust-estimator family of Figure 2, whose domains include corruption-tolerant bandits. Its missing leaves are a corruption-robust policy and a regret theorem, which no declaration covers, and the corruption-tolerant setting is listed in BanditRLwiki with its source audit pending. The overlay states what the extension inherits, what it might share, and what it still owes, and leaves the judgment of its value to the reader.

4 The Reusable Mathematical Spine

Many proofs need the same statement with different objects plugged in. BanditRLlib states such a result once, with explicit hypotheses, and lets each development supply its own objects. Four of these shared contracts carry most of the reuse in this paper.

Interaction law. Adaptive data are not a table of samples waiting to be read. The algorithm creates the data-generating process, and every regret argument reasons about that process. The library constructs it once: from an algorithm, a map from histories to action distributions, and a Markov kernel of reward laws per arm, it builds the history as a measurable space and the law of the history as a measure, with the measurability facts that adaptive arguments need. Upper-bound developments define their trajectories with this construction, and the change-of-measure arguments of lower bounds compare two such laws.

Concentration contract. Most confidence arguments need only one inequality about a statistic X under a law μ at a fixed tilt t,

$$\mathbb{E}_\mu\big[e^{tX}\big] \le e^b,$$

with a budget b that the proof then converts into a confidence radius. The library stores this weakest form rather than a full moment-generating-function bound, because a truncation level, a bias, a second moment, and an admissible range of tilts are model-specific facts that each development proves for its own statistic. The contract is stated for a Markov kernel, so that the conditional law of an observation given the past is an instance, and it retains exponential integrability at every tilt so that conditional laws compose. Two entries feed it: an independent-sum entry that adds budgets over independent summands, and a conditional entry that controls an adaptively selected mixture given the past. The two are not interchangeable, and which one a development uses is itself information about its sampling model.

Regret criteria. Different developments bound different quantities, and the atlas compares like with like. For finite stochastic bandits with pull counts $N_a(T)$ and gaps Δ_a, the *pseudo-regret* is $\sum_a \Delta_a N_a(T)$. When the reward of round t is drawn from the selected arm's law given the past, so that $\mathbb{E}[Y_t \mid \mathcal{F}_{t-1}, A_t] = \mu_{A_t}$, the expected *realized regret* equals the expected pseudo-regret at every horizon. A recommendation is judged by its *simple regret*, an approximation oracle by an *oracle-relative regret* that can be negative, and an RL policy by its *episode pseudo-regret*. Each is its own random quantity or expectation, and the coverage table names which one a development bounds.

History divergence. The lower-bound counterpart of the concentration contract is one identity on the interaction law. For a finite horizon n, one possibly randomized non-anticipating policy π shared by two environments, and two Markov reward kernels ν, ν',

$$D_{\mathrm{KL}}\big(P_\nu^\pi \,\big\|\, P_{\nu'}^\pi\big) = \sum_a \mathbb{E}_\nu^\pi\big[N_a(n)\big]\, D_{\mathrm{KL}}(\nu_a \,\|\, \nu_a'),$$

with divergences valued in $[0, \infty]$ and no absolute-continuity assumption. One common interaction semantics and one divergence decomposition serve every lower bound. What changes between bounds is the choice of hard environments and the change-of-measure step (Section 5.3).

5 What the graph reveals across settings

5.1 Heavy-tailed and causal bandits: one concentration layer

The policies and objectives differ, but their concentration arguments admit the same formal contract. In the heavy-tailed setting (Bubeck et al., 2013), K arms have stationary independent reward streams with only a raw moment bound $\mathbb{E}|X_a|^{1+\epsilon} \le u$, $0 < \epsilon \le 1$, the policy is the source's truncated-mean UCB rule with its factor-four confidence radius, and the criterion is expected pseudo-regret over T rounds. In the causal setting (Lattimore et al., 2016), an action intervenes on a finite directed acyclic graph with a binary reward node, the learner knows each action's law P_a of the reward's parents but not the reward table, it spends a budget of T samples by a fixed design η and estimates means with importance weights P_a/Q truncated at a level B, where $Q = \sum_a \eta_a P_a$, and the criterion is the expected simple regret of one recommendation.

What is shared. Both analyses control the fluctuation of a centered, truncated statistic, and the library provides that step as two lemmas. The *centering lemma*: for a measurable Y with $|Y| \le B$ and $\mathbb{E}\,Y^2 \le v$ and a tilt with $|t| \cdot 2B \le 1$, the centered variable $Y - \mathbb{E}Y$ has budget $t^2 v$. The *independent-sum lemma*: independent summands with budgets b_i have a sum with budget $\sum_i b_i$. Both were written during the heavy-tailed development, and the causal proof calls them by name (Figure 3). A later paper that changes the estimator but satisfies the same contract can reuse the downstream proof, and a paper whose statistic fails the contract exposes exactly where new mathematics is needed.

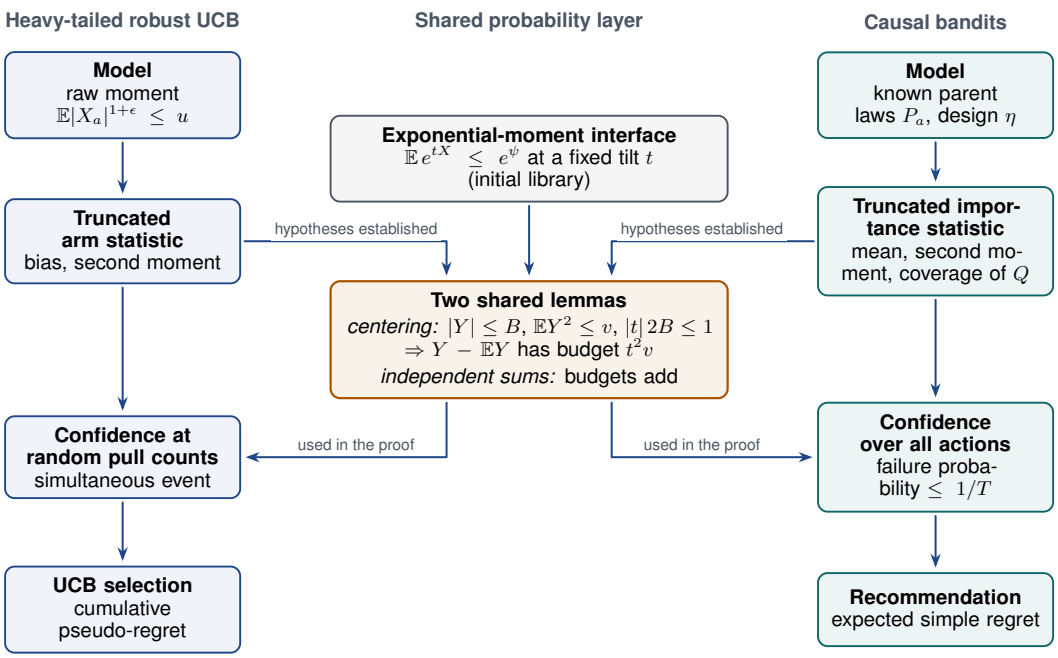

Figure 3: One shared probability layer, two proofs. Solid arrows are compiled dependencies at this release. The two shared lemmas are the bounded-centering and independent-sum exponential-moment lemmas of the heavy-tailed module, and the contract they instantiate is the fixed-tilt exponential-moment bound of the initial library. Each proof establishes the lemmas' hypotheses in its own model, and the arguments after the shared layer, adaptive pull counts on one side and a recommendation under an intervention model on the other, do not transfer.

Where the settings diverge. The heavy-tailed proof derives the bias and second moment of its truncated statistic from the raw-moment assumption, turns fixed-count bounds into confidence at the random pull count of each arm through a simultaneous event, and combines the confidence with the UCB rule and the pull-count decomposition. The causal proof constructs the joint law of one sample under an intervention in topological order, derives the mean and second moment of the importance statistic under a support-coverage condition on Q, obtains a confidence event simultaneous over actions with failure probability at most $1/T$, and converts it into a simple-regret bound. Neither the UCB rule with its pull counts nor the design with its recommendation rule crosses between the two, and that boundary is what the contract fixes. Two later developments show its granularity. Combinatorial bandits with triggered arms use the conditional entry, because their observations depend on earlier actions, and the clipped estimator of Section 5.4 calls the same two lemmas and adds one of its own.

5.2 HOO: MODEL ONLY WHAT THE ALGORITHM VISITS

A general interface can quietly narrow a theorem. Hierarchical optimistic optimization (Bubeck et al., 2011) searches an arm space $\mathcal{X}$ through a binary tree of cells whose diameters shrink geometrically with depth, and it only ever pulls one fixed representative $r(v)$ per cell v. The source assumes a reward law M_x on $[0,1]$ with mean $f(x)$ for every arm and nothing about how M_x varies with x. A first formalization of the regret theorem modeled rewards as a Markov kernel on the whole arm space, which requires $x \mapsto M_x$ to be measurable. That premise belongs to the interface, not to the theorem. The adapter replaces it by a construction on the objects the algorithm visits: let V be the countable set of tree nodes with the discrete σ-algebra and set $K(v,\cdot) = M_{r(v)}$. Every subset of V is measurable, so any assignment of reward laws to nodes is a measurable Markov kernel, and K exists with no assumption on $x \mapsto M_x$. The construction needs one globally fixed countable set of representatives, which the covering provides, and not merely the fact that each run samples countably many arms. The library builds the trajectory law from K, proves that K agrees

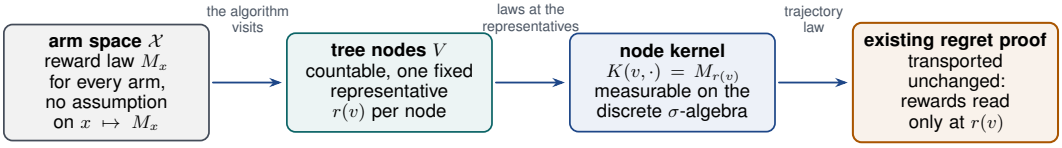

Figure 4: The HOO adapter. The source assumes a reward law for every arm and nothing about how the laws vary. The algorithm only visits the countable node set with its fixed representatives, so the reward laws at representatives form a measurable kernel on the discrete σ-algebra, and the existing regret proof is transported to that kernel without the global measurability premise that a generic interface had added.



Figure 5: The three layers of a lower-bound proof. The first two are shared library objects. The third is what each bound constructs: the minimax bound perturbs one of m arms of a Gaussian base environment, the instance-dependent bound uses one-arm alternatives, and the high-probability bound uses a correlated clipped-Gaussian path construction.

with a global kernel on the representatives whenever one exists, and transports the existing regret analysis, which evaluates rewards only at representatives (Figure 4). The transported theorem keeps the source's assumptions, $f^* = \sup f$, the weak Lipschitz condition, and bounded rewards, and gives, for every d above the near-optimality dimension, a constant γ with expected regret at most $\gamma N^{(d+1)/(d+2)}(\log \max\{N, 2\})^{1/(d+2)}$ for every horizon N, with no measurability of $x \mapsto M_x$, no compactness, and no attained optimum (Appendix D.2). The lesson is a library principle: model only what the algorithm needs, and when a generic abstraction narrows a source theorem, adapt the abstraction rather than strengthen the theorem. Two amendments of a different kind, to the displayed covering recursion and to the logarithmic term at the smallest horizon, are recorded beside the source (Appendix B).

5.3 Lower bounds: one interaction semantics, different hard alternatives

Lower-bound proofs look different because their hard instances differ, but their first two layers are the same. (1) The interaction history under a policy and an environment is constructed by the interaction law. (2) The divergence between two such histories decomposes into per-arm terms by the identity of Section 4. (3) Only the third layer is specific to a bound: the hard environments, the event compared across them, and the change-of-measure inequality (Figure 5). BanditRLlib shares the first two layers and leaves the third to each development. At this release the identity is called by the three lower-bound developments of the textbook (Lattimore & Szepesvári, 2020). The minimax development uses it for the unit-Gaussian bound in the form proved: for $k = m + 1$ arms and any horizon $n \geq m$, every algorithm has expected pseudo-regret at least $\frac{1}{27}\sqrt{mn}$ in the base environment or in one of m perturbations. The instance-dependent development uses it for the one-arm change-of-measure step behind the asymptotic bound for consistent policies, and the high-probability development for the bound on the probability that the random pseudo-regret is large. What each development adds above the shared layers is exactly its own idea.

5.4 A clipped estimator: an inherited route plus one lemma

The cleanest composition in the library is an extension of the heavy-tailed route. Hard truncation is replaced by clipping to $[-B, B]$, and the consumed observations may be corrupted with a pathwise

budget C on the total absolute corruption. Clipping is 1-Lipschitz, so the clipped prefix means of clean and corrupted samples over N observations differ by at most C/N. Within the formal route represented here, this perturbation lemma is the only new ingredient. The clean analysis of the clipped mean reuses the centering and independent-sum lemmas, and the resulting confidence statement enlarges the radius by C/N at an outcome-dependent count $0 < N \leq t$, along an arbitrary action trace, including a policy driven by the corrupted observations. The graph shows the extension as inherited structure plus one new node. The extension is an estimator guarantee, and no corruption-robust policy or regret theorem is built on it yet. The same graph shows what such a theorem would still need: a selection rule whose analysis consumes the enlarged radius, and a regret decomposition that charges the corruption budget, neither of which exists as a declaration.

5.5 Musical Chairs: where nothing is shared yet

Not everything shares a core. The multi-player development (Rosenski et al., 2016) proves its aggregate pseudo-regret bound end to end, but its coordination argument rests on collision counts and rankings for which the shared foundations offered nothing, and it calls no declaration of the initial library. The graph shows it as a separate component. The mechanism hypergraph records the corresponding mechanism, many learners coordinating through collisions, as a candidate family with no Lean substrate yet, which is the honest state: a conceptual correspondence and a named obligation, not a shared node. A shared coordination interface would have to make the collision-count process and the ranking from collision-free rewards objects of the interaction law, and until it exists the development stays a separate component of the graph.

6 Coverage, Frontier, and trustworthy growth

6.1 What is formalized

Table 1 shows the shape of the coverage at this release, grouped as the textbook backbone, the five completed setting extensions, finite-horizon RL, and the Frontier routes. The full table with criteria, statuses, and foundations is Appendix C. The RL book reaches a learning algorithm. On finite Markov decision processes with known rewards in $[0, 1]$, a Hoeffding-type UCB-VI development (Azar et al., 2017) constructs the algorithm's own trajectory law over K episodes, with transition estimates from exact counts, a clipped optimistic planner, and a strict-prefix measurable policy, and proves that with probability at least $1 - \delta$ the cumulative episode pseudo-regret is at most $20H\sqrt{H}\,L\sqrt{SAK} + 250H^2S^2A\,L^2$, where L is a logarithmic factor in S, A, H, K, and $1/\delta$, with the matching expectation bound larger by $KH\delta$. Bernstein-type variance-aware rates, stochastic rewards, and posterior sampling are named leaves of the Frontier. The Frontier keeps two kinds of leaf apart. Formalization leaves, such as these RL rates and the three partial routes, are open in the library and closed in the literature. Literature leaves, such as the textbook's distributional-regret conjecture and regret minimization with unknown heavy-tail parameters, are open mathematics, and the library records them without claiming progress on them.

6.2 Reuse first, promote on a second use

The library grows by a simple rule. A contributor searches Mathlib and BanditRLlib before writing a lemma. A lemma with one real use stays local to its route. When a genuinely different proof needs the same mathematical contract, the lemma moves to the shared layer with an explicit adapter, and the reference records show the move. Conceptual similarities live in the dashed hypergraph until they deserve a formal abstraction. The two concentration lemmas of Section 5.1 are the current test case: written for heavy-tailed rewards, they now have a second and a third proof and are due to move. The reference records make the rule checkable: a declaration called from the modules of two developments other than its own is the signal to promote it, and Appendix G reports the compiled reference counts per development that show which files those are. The counts locate reuse and do not measure novelty.

Table 1: Coverage map at this release. Criteria, exact statuses, and the foundations each development uses are listed in Appendix C.

Branch	Content	State
Textbook backbone	explore-then-commit, UCB, OFUL, Thompson sampling, EXP3, Tsallis-FTRL; information-theoretic tools; minimax, instance-dependent, and high-probability lower bounds	complete; two high-probability claims specialized or amended
Setting extensions	heavy-tailed UCB; HOO; CUCB with triggered arms; causal bandits; Musical Chairs	complete, each from the algorithm's own sampling law
Finite-horizon RL	Bellman optimality; occupancy identity; optimism certificate; Hoeffding UCB-VI regret bound	complete for known rewards; Bernstein rate open
Frontier routes	stochastic-gradient bandit Theorem 2; delayed best-of-both-worlds; succinct lower bound	partial, each with a named open leaf

6.3 Keeping a growing library trustworthy

Lean checks that a proof establishes the submitted proposition. Whether the proposition says what the cited theorem says is a judgment, and the atlas would gain apparent reuse if assumptions, objectives, algorithms, or quantifier order were quietly changed. The protocol therefore runs a semantic round trip on every source-facing theorem: the formal statement is handed, without the source, to a second party who reconstructs the mathematical claim with all of its assumptions, and a third party compares that reconstruction with the source and records every difference. The invariant is that the source statement, the reconstruction, the Lean statement, its assumptions, and its position in the graph must be reconciled, and any discrepancy is explicit before the node is trusted. The claim is fixed and the proof strategy is free: contributors may change a decomposition, generalize a shared lemma, or retract an intermediate definition, while changes to assumptions or conclusions must be disclosed. The HOO premise of Section 5.2 was found by this round trip. Differences from a source are recorded beside the formal result and classified as a counterexample certificate, an amendment, an adapter of our own interface, a specialization, or an open obligation (Appendix B). One example: the printed bound $32 \log T + 5$ on inferior-arm pulls in the heavy-tailed source fails for an admissible two-arm instance, Lean proves the failure by a symbolic count, and a bound with a larger coefficient is proved for the unchanged policy. The mechanical checks (compilation under the pinned toolchain, checked instantiations, reachability, placeholder and axiom audits, regenerated indices) and the review records are described in Appendix F. The reviewing roles were filled by isolated AI instances under the authors' supervision.

7 Discussion and conclusion

A field that produces results faster also needs infrastructure for digesting them faster. BanditRLlib offers one form of it: a trusted classical backbone in Lean, explanations attached to exact statements, a compiler-derived graph of what depends on what, a conceptual hypergraph of what might transfer and where it fails, and an atlas that says which assumptions a result was proved under and which leaves remain open. On this substrate the anatomy of a proof becomes inspectable. Two settings that meet at one concentration lemma, a probability model rebuilt on the objects an algorithm visits, three lower bounds on one interaction semantics, an estimator extended by a single lemma, and a coordination argument that shares nothing yet are visible as structure rather than asserted in prose. A reader can trace a textbook theorem to its Lean dependencies, a researcher can compare assumptions and proof branches across settings, and a contributor can overlay a new result on the graph to see what it inherits, what it recombines, and what it still owes.

The claims of this paper are limited to the formalized developments of Appendix C. Correspondence between formal statements and their sources remains an author-assessed judgment supported by documented AI-assisted comparisons, and the effect of the views on readers is a question for future evaluation (Appendix A). The Frontier names the leaves where the next mathematics enters.

AI USE STATEMENT

Large language models assisted with writing and editing this paper and with the Lean formalization, including proof search and repair. Isolated AI instances also performed the source comparisons recorded with the artifact (Appendix F) under the authors' supervision, and those records are not independent expert validation. The authors reviewed the mathematical claims, the formal statements, and the explanations, and take responsibility for the content.

REFERENCES

Mark Miles Adams. Proof auditing formalised mathematics. *Journal of Formalized Reasoning*, 9 (1):3–32, January 2016. doi: 10.6092/issn.1972-5787/4576. URL https://jfr.unibo.it/article/view/4576.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pp. 263–272, 2017.

Dorian Baudry, Emmeran Johnson, Simon Vary, Ciara Pike-Burke, and Patrick Rebeschini. Does stochastic gradient really succeed for bandits? In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 113544–113611. Curran Associates, Inc., 2025. doi: 10.52202/085713-3787. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a4e683f0ce6b91e7fbdae9d32642d88f-Paper-Conference.pdf.

Lars Becker, María Inés de Frutos-Fernández, Leo Diedering, Floris van Doorn, Sébastien Gouëzel, Asgar Jamneshan, Evgenia Karunus, Edward van de Meent, Pietro Monticone, Jasper Mulder-Sohn, Jim Portegies, Joris Roos, Michael Rothgang, Rajula Srivastava, James Sundstrom, Jeremy Tan, and Christoph Thiele. A blueprint for the formalization of Carleson's theorem on convergence of Fourier series, 2024. URL https://arxiv.org/abs/2405.06423.

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. X-armed bandits. *Journal of Machine Learning Research*, 12:1655–1695, 2011.

Sébastien Bubeck, Nicolò Cesa-Bianchi, and Gábor Lugosi. Bandits with heavy tail. *IEEE Transactions on Information Theory*, 59(11):7711–7717, 2013.

Rafael Cabral, Tuan Manh Do, Xuejun Yu, Wai Ming Tai, Zijin Feng, and Xin Shen. ProofFlow: A dependency graph approach to faithful proof autoformalization. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5fce5198dbf92d5ff45f74504431e2ff-Abstract-Conference.html.

Guoxin Chen, Jing Wu, Xinjie Chen, Wayne Xin Zhao, Ruihua Song, Chengxi Li, Kai Fan, Dayiheng Liu, and Minpeng Liao. ReForm: Reflective autoformalization with prospective bounded sequence optimization, 2026. URL https://arxiv.org/abs/2510.24592.

Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *Journal of Machine Learning Research*, 17(50):1–33, 2016.

Rob Cornish, Iacopo Ghinassi, Po-Hung Yeh, Shuqi Liu, Qiyuan Xu, Haoxuan Yin, Dominik Wagner, Wenda Li, Yee Whye Teh, and Luke Ong. FaithformBench: Benchmarking faithfulness of mathematical chain-of-thought autoformalisation, August 2026. URL https://arxiv.org/abs/2608.10916.

Leonardo de Moura and Sebastian Ullrich. The lean 4 theorem prover and programming language. In *Automated Deduction–CADE 28*, pp. 625–635. Springer, 2021.

Yaël Dillies, Terence Tao, et al. Formalization of the polynomial Freiman–Ruzsa conjecture of Marton. Lean 4 formalization project, 2023. URL https://github.com/teorth/pfr.

Yuming Feng, Frederick Pu, One An, Osbert Bastani, Li Zhang, Jiani Huang, Xujie Si, and Ziyang Li. Theory-scale auto-formalization of logics for computer science, June 2026. URL https://arxiv.org/abs/2606.26525.

Yiming Fu, Peixuan Liu, Zichen Wang, and Kun Yuan. Proof-Refactor: Refactoring generated formal proofs into modular artifacts, 2026. URL https://arxiv.org/abs/2606.03743.

Fabian Gloeckle, Ahmad Rammal, Charles Arnal, Remi Munos, Vivien Cabannes, Gabriel Synnaeve, and Amaury Hayat. Automatic textbook formalization, April 2026. URL https://arxiv.org/abs/2604.03071.

Zhengtao Gui, Sheng Yang, and Zhouxing Shi. Evaluating the robustness of proof autoformalization in Lean 4, 2026. URL https://arxiv.org/abs/2606.14867.

Finnian Lattimore, Tor Lattimore, and Mark D. Reid. Causal bandits: Learning good interventions via causal inference. In *Advances in Neural Information Processing Systems 29*, 2016.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

Xiaoyang Liu, Tao Zhu, Zineng Dong, Yuntian Liu, Qingfeng Guo, Zhaoxuan Liu, Yu Chen, and Tao Luo. ASSESS: A semantic and structural evaluation framework for statement similarity. In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10008659.

Yuntian Liu, Tao Zhu, Xiaoyang Liu, Yu Chen, Zhaoxuan Liu, Qingfeng Guo, Jiashuo Zhang, Kangjie Bao, and Tao Luo. Generalized tree edit distance (GTED): A faithful evaluation metric for statement autoformalization, 2025. URL https://arxiv.org/abs/2507.07399.

Jianqiao Lu, Yingjia Wan, Yinya Huang, Jing Xiong, Zhengying Liu, and Zhijiang Guo. FormalAlign: Automated alignment evaluation for autoformalization. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=B5RrIFMqbe.

Patrick Massot. leanblueprint: A plasTeX plugin to build formalization blueprints for Lean. GitHub repository, 2020. URL https://github.com/PatrickMassot/leanblueprint.

Jordan Meadows, Lan Zhang, and Andre Freitas. FormalScience: Scalable human-in-the-loop autoformalisation of science with agentic code generation in Lean, 2026. URL https://arxiv.org/abs/2604.23002.

Lazar Milikic, Simon Guilloud, Khanh Nguyen, and Viktor Kuncak. Leanflow: A case study in workflow-driven lean autoformalization, 2026. URL https://arxiv.org/abs/2607.20503.

Marcus J. Min, Mike He, Zhaoyu Li, Zixuan Yi, Sharad Malik, Aarti Gupta, Xujie Si, and Osbert Bastani. Theory-level autoformalization: From isolated statements to unified formal knowledge bases, 2026. URL https://arxiv.org/abs/2607.13292.

Noor Islam S. Mohammad and Tamim Sheikh. The faithfulness gap: Certifying semantic equivalence between natural-language and formal mathematical statements, 2026. URL https://arxiv.org/abs/2606.16541.

Wesley Pegden. (Auto)formalization Is Supposed to Be Easy: Trellis process semantics for spelling out rigorous proofs, 2026. URL https://arxiv.org/abs/2606.09674.

Auguste Poiroux, Gail Weiss, Viktor Kunčak, and Antoine Bosselut. Reliable evaluation and benchmarks for statement autoformalization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 17947–17969. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.907. URL https://aclanthology.org/2025.emnlp-main.907/.

Jonathan Rosenski, Ohad Shamir, and Liran Szlak. Multi-player bandits – a musical chairs approach. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *PMLR*, pp. 155–163, 2016.

Ofir Schlisselberg, Tal Lancewicki, Peter Auer, and Yishay Mansour. Improved best-of-both-worlds regret for bandits with delayed feedback. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 2035–2072. Curran Associates, Inc., 2025. doi: 10.52202/085713-0075. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/02f0ac0a323dc17d964d4bbf8a62e01b-Paper-Conference.pdf.

Peter Scholze. Liquid tensor experiment. *Experimental Mathematics*, 31(2):349–354, 2022. doi: 10.1080/10586458.2021.1926016.

Aleksandrs Slivkins. Introduction to multi-armed bandits. *Foundations and Trends in Machine Learning*, 12(1–2):1–286, 2019.

Sho Sonoda, Kazumi Kasaura, Yuma Mizuno, Kei Tsukamoto, and Naoto Onda. Lean formalization of generalization error bound by rademacher complexity. *arXiv preprint arXiv:2503.19605*, 2025. URL https://arxiv.org/abs/2503.19605.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, second edition, 2018.

The mathlib Community. The Lean Mathematical Library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP 2020, pp. 367–381, New Orleans, LA, USA, January 2020. ACM. doi: 10.1145/3372885.3373824. URL https://doi.org/10.1145/3372885.3373824.

Dingzirui Wang, Xuanliang Zhang, Keyan Xu, Qingfu Zhu, and Wanxiang Che. FormalTCS: Benchmarking end-to-end frontier formal theoretical computer science research of large language models, August 2026a. URL https://arxiv.org/abs/2608.20153.

Zichen Wang, Wanli Ma, Zhenyu Ming, Gong Zhang, Kun Yuan, and Zaiwen Wen. M2F: Automated formalization of mathematical literature at scale, 2026b. URL https://arxiv.org/abs/2602.17016.

Jiayi Wu, Robert Joseph George, and Anima Anandkumar. ITPEval: Benchmarking formal translation across interactive theorem provers, July 2026. URL https://arxiv.org/abs/2607.19407.

Guo Zeng and Jean Honorio. A novel general framework for sharp lower bounds in succinct stochastic bandits. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 26678–26699. Curran Associates, Inc., 2025. doi: 10.52202/085713-0898. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/26407588dfb08e48c459c074ab6adb7d-Paper-Conference.pdf.

Ke Zhang, Patricio Gallardo Candela, Sudhir Murthy, Yi Xie, Zhi Wang, and Maziar Raissi. Beyond compilation: Evaluating faithful natural-language-to-Lean statement formalization, June 2026a. URL https://arxiv.org/abs/2606.31002.

Shangtong Zhang. Towards formalizing reinforcement learning theory: A Robbins–Siegmund approach. *arXiv preprint arXiv:2511.03618*, 2025. URL https://arxiv.org/abs/2511.03618.

Yuanhe Zhang, Jason D. Lee, and Fanghui Liu. AI4SLT: Empirical processes in Lean 4 for formal statistical learning theory. In *International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=dfqmQ9WhCP.

Yuanhe Zhang, Yuekai Sun, Taiji Suzuki, Jason D. Lee, and Fanghui Liu. Leanmarathon: Toward reliable ai co-mathematicians through long-horizon lean autoformalization, 2026c. URL https://arxiv.org/abs/2606.05400.

Thomas Zhu, Pietro Monticone, Jeremy Avigad, and Sean Welleck. LeanArchitect: Automating blueprint generation for humans and AI, 2026. URL https://arxiv.org/abs/2601.22554.

A Limitations, broader impact, and future work

Coverage. The claims in this paper are limited to the formalized developments listed in Table 4. BanditRLwiki, the technique map, and the mechanism hypergraph also contain conceptual and literature-only entries, which are labeled separately and are not formalization claims. The focal developments are a documented selection, not a sample of the literature, so the differences of Appendix B describe these sources only.

Correspondence with sources. Lean checks that each proof establishes its formal statement. Whether a formal statement matches its source was judged by the authors with AI-assisted comparisons (Appendix F). No external expert has re-judged a development, and such a review can miss a mismatch. Kernel verification and this judgment are different kinds of evidence. Where amended guarantees retain the asymptotic order of a source, their changed conditions, constants, and parameters are still stated, and retaining a rate does not make an amended statement the source's statement.

Reuse figures. The reference counts of Appendix G are extracted from compiled proofs at one release and include generated declarations. They locate reuse and do not measure use frequency, effort saved, or novelty, and a refactoring can change them without changing any mathematics.

Reader views. The workspace, the Lean Graph, the Functor Hypergraph, and BanditRLwiki with its Frontier are implemented at this release. Their effect on reader understanding and on development effort is a question for future evaluation. The graph's progressive disclosure compresses the presentation; no compression of proofs is claimed. The hypergraph's correspondences are conceptual overlays, not theorem dependencies.

Broader impact. The direct effect of a source-aligned library is on how theory is read, reused, and corrected: natural-language explanations make formal results accessible, shared lemmas reduce duplicated effort, and explicit assumptions and recorded differences make review more transparent. The graph supplies evidence for expert judgment and does not automate a novelty ranking. Amendments to published derivations are presented beside the unchanged sources and, where they concern constants or displayed steps, leave the sources' main results intact.

Future work. Open formalization leaves, new settings, and literature-open questions are represented in the Frontier rather than only listed here. The first direction is the shared layer: moving lemmas with several proofs out of the developments that first proved them, and adding interfaces for coordination and delayed feedback. The second is coverage: completing the partial routes, extending the textbook-aligned chapters, and reaching the Bernstein-type and stochastic-reward RL rates. The third is readers: evaluating whether the views help people locate assumptions and reusable results. Independent expert review of the source-alignment judgments remains a quality-assurance step for all three.

B Recorded differences between sources and formal developments

Tables 2 and 3 list the recorded differences between a source and its formal development, one difference per row, classified by the mathematical relation between the formal statement and the source and by the strongest evidence that exists for it. Kinds: C, counterexample certificate; A, amendment to a displayed derivation or statement; F, formalization adapter (our interface, not the source); S, specialization of the source statement by explicit constants or an added hypothesis; O, open obligation or unresolved source step. Kinds C and A record a discrepancy in the source, kind F records a repair to our own interface, kind S narrows the formal statement, and kind O is an unfinished obligation on our side, not an error in the source. The rows are therefore not a defect count.

C Full coverage table

Table 4 lists every formalized development with the performance criterion its guarantee concerns, the status of the formal development, and the foundations of Section 4 its proof uses. "Complete" means

Table 2: Recorded differences for the sources of the five focal developments.

Source	Kind	Difference or obligation	Evidence established	Effect on scope
Heavy-tailed (Bubeck et al., 2013)	C	printed bound $32 \log T + 5$ on inferior-arm pulls fails for an admissible two-arm instance	Lean proves the failure at $T = 2^{50}$ by a symbolic count; a bound with a larger coefficient proved for the unchanged policy	rate and other estimators unaffected
HOO (Bubeck et al., 2011)	F	our first formalization required a measurable kernel on the whole arm space	countable-node kernel on the discrete σ-algebra; theorem reproved without the premise	source assumption set restored
HOO (Bubeck et al., 2011)	A	covering recursion of equation (2) writes the right child at the parent's depth; Algorithm 1 prints inconsistent indices for the root's children	implemented recursion recorded beside the printed one; both root children initialized to infinite indices	algorithm display
HOO (Bubeck et al., 2011)	A	$\log N$ in the confidence term and the bound vanishes at $N = 1$	bound with $\log \max\{N, 2\}$ proved; agrees with the printed form for $N \geq 2$	logarithmic term at the smallest horizon
CUCB (Chen et al., 2016)	A	threshold rescaling by trigger probabilities is not an identity across the two analysis branches	normalized-counter charge proved; action rule unchanged	branch constants differ
CUCB (Chen et al., 2016)	S	the formal statement assumes that the smoothness modulus attains every positive gap, so that its inverse exists on the positive-gap interval	hypothesis stated in the source model; not implied by strict monotonicity alone	validity range
Causal (Lattimore et al., 2016)	A	final constant counts one of two confidence radii	bound with $2\sqrt{2} + 7$ proved; smaller constants not ruled out	constant
Causal (Lattimore et al., 2016)	A	displayed probability bounds the good event where the bad one is meant	$\Pr(\text{bad}) \leq 1/T$ proved by a union bound	none
Causal (Lattimore et al., 2016)	A	printed parallel design has total mass $3/2$	normalized design with mass $1 - D$; cost bound requires strict rarity	design and a stated condition
Musical Chairs (Rosenski et al., 2016)	F	coordination theorem takes a correct candidate set as input	statistical producer proved; exploration budget enlarged for an all-player union bound	tuning of the exploration length

that the algorithm's actual sampling law is constructed and the stated guarantee is proved from it without an assumed tail bound. "Amended" means that the proved statement differs from the printed one in a recorded way, "specialized" that it is proved with explicit constants or an added hypothesis, and "partial" names the next obligation. The criteria differ across rows: causal bandits bound an expected simple regret, combinatorial bandits a regret against an approximation oracle that can be negative, and Musical Chairs an expected aggregate pseudo-regret at a fixed confidence level δ, with a term $\delta n H$ that is linear in the horizon.

D THE FIVE FOCAL DEVELOPMENTS

Each development is described in the same order: the problem and the algorithm, the library mathematics it reuses, the obligations it must establish itself, the guarantee proved, the recorded differences from the source, and the concrete instance that exercises the actual policy. Reference counts are those of Appendix G.

Table 3: Recorded differences for the partial routes and the textbook chapters.

Source	Kind	Difference or obligation	Evidence established	Effect on scope
Delayed (Schlisselberg et al., 2025)	F	two timing conventions for available feedback	both sets defined; partition and one-based bridge proved	none
Delayed (Schlisselberg et al., 2025)	O	confidence width used against its displayed monotonicity	consequence proved conditionally on the used direction	theorems remain open
Succinct (Zeng & Honorio, 2025)	O	dual quantity is a supremum over an unbounded set when a vector is orthogonal to every atom	unboundedness proved in Lean; spanning, domain restriction, or extended-real codomain are distinct repairs	Thm 3.8 remains open
Succinct (Zeng & Honorio, 2025)	O	proof uses two sample-size regimes the statement omits	recorded as a review finding	separate obligation
SGB (Baudry et al., 2025)	O	a Theorem-4 appendix step conditions on $q \geq c$ and $q < c$ alternately	recorded against the source; not refuted, not proved	Thms 1–2 unaffected
Textbook Ch. 17 (Lattimore & Szepesvári, 2020)	A	Claim 17.6 proved with $T_i \leq n/2$ rather than as printed	amended claim proved	event condition
Textbook Ch. 17 (Lattimore & Szepesvári, 2020)	S	Thm 17.4 instantiated with explicit constants ($\delta \leq 1/32$, $c = 1/160$, $C = 64$)	specialized statement proved	validity range

D.1 Heavy-tailed rewards: robust UCB

Problem and algorithm. A finite arm set with stationary independent reward streams and a known $u > 0$ with $\mathbb{E}|X_a|^{1+\epsilon} \leq u$ for every arm, $0 < \epsilon \leq 1$ (Bubeck et al., 2013). The assumption is a raw moment bound, so rewards may be unbounded and negative; it is not replaced by a centered moment, and integrability is derived from it. The policy is the source's truncated-mean UCB rule, including its factor-four confidence radius, with round-robin initialization as the tie convention for unpulled arms. The truncated estimator depends on both the sample index and the decision round; substituting an independent fixed sample for the adaptive observations would change the theorem.

Reusable mathematics. The fixed-tilt exponential-moment contract of the initial library and the two lemmas of Section 5.1, which were written here; the pull-count decomposition of pseudo-regret; and the bridge from fixed-count bounds to confidence at the random pull count $N_a(t)$, which requires a simultaneous event over counts rather than the substitution $n = N_a(t)$.

Setting-specific obligations. The bias and the second moment of the truncated statistic, derived from the raw-moment assumption; signed concentration; the selection and count bounds of the UCB rule.

Guarantee. Let $\Delta_a = \mu^* - \mu_a$ and let $R_T = \sum_{t \leq T} \Delta_{A_t}$ be the pseudo-regret. For every horizon T,

$$\mathbb{E}[R_T] \leq \sum_{a\colon \Delta_a > 0} \Delta_a \big(A_a(T) + 5\big), \qquad A_a(T) = \frac{2 \log \max\{T, 1\}}{\big(\Delta_a / (8u^{1/(1+\epsilon)})\big)^{(1+\epsilon)/\epsilon}},$$

so that each suboptimal arm contributes a term of order $u^{1/\epsilon} \Delta_a^{-1/\epsilon} \log T$ with an ϵ-dependent constant. For $\epsilon = u = \Delta_a = 1$ the bound on inferior-arm pulls is $128 \log T + 5$.

Source differences. The printed bound $32 \log T + 5$ on inferior-arm pulls is false for an admissible instance: two deterministic arms with rewards 0 and -1, $\epsilon = u = 1$, $T = 2^{50}$, and the implemented

Table 4: Source-aligned developments in BanditRLlib at this release. The performance criterion is the quantity each guarantee bounds, stated per row as a random quantity or as its expectation. Statuses are generated from the recorded development state. Full source locators and Lean identifiers are in the artifact.

Source and setting	Performance criterion	Status	Foundations used
Bandit textbook, Part IV, Ch. 13–17 (Lattimore & Szepesvári, 2020)	minimax and instance-dependent lower bounds on expected pseudo-regret	complete for the listed theorems; Claim 17.6 amended; Thm 17.4 specialized with explicit constants	history divergence identity; change of measure
Heavy-tailed robust UCB (Bubeck et al., 2013), Thm 1	expected pseudo-regret, raw $(1+\epsilon)$-moment	complete with a larger coefficient; printed coefficient refuted by a finite certificate	fixed-tilt exponential moments; pull-count decomposition
HOO (Bubeck et al., 2011), Alg. 1, Thm 6	expected regret over a covering tree	complete under the source's assumptions; covering recursion and small-horizon logarithm amended	adaptive conditional concentration; trajectory law
CUCB with triggered arms (Chen et al., 2016), Alg. 1, Thms 1–2	$H\alpha\beta\,\mathrm{OPT} - \mathbb{E}\sum_t Y_t$ against an (α, β) oracle	complete with a normalized-counter analysis; inverse of the smoothness modulus assumed	conditional exponential moments; layer-cake integration
Causal bandits (Lattimore et al., 2016), Alg. 2, Thm 3, Prop. 4	expected simple regret of the recommendation	complete with constant $2\sqrt{2}+7$ and a normalized parallel design	centering and independent-sum lemmas; finite argmax
Musical Chairs (Rosenski et al., 2016), Algs. 1–2, Thm 1	expected aggregate pseudo-regret at fixed δ	complete end to end, with an enlarged exploration budget	none from the initial library
Stochastic-gradient bandit (Baudry et al., 2025), Thm 1, Cor. 1	expected regret, two arms, independent rewards	complete	interaction law
Stochastic-gradient bandit, Thm 2	expected regret through a phase event	partial: latent-stream identification and missing-pull charge; trigger event open	trajectory-law bridges
Delayed best-of-both-worlds (Schlisselberg et al., 2025), Thms 4.1/5.1	regret of one algorithm in two regimes	partial: delay accounting, processing, elimination, union bound; confidence evolution open	same-algorithm multi-regime contract
Succinct lower bounds (Zeng & Honorio, 2025), Thm 3.8	lower bound on regret	partial: Defs. 3.1–3.3, Lemmas 3.1–3.4; a spanning premise and two sample-size regimes open	none
Finite-horizon RL, finite MDP, known rewards (Azar et al., 2017)	cumulative episode pseudo-regret over K episodes, high probability and expectation	complete: Hoeffding UCB-VI bound $20H\sqrt{HL}\sqrt{SAK} + 250H^2S^2AL^2$ with probability $1-\delta$; Bernstein rate and stochastic rewards open	Bellman optimality; occupancy identity; optimism certificate; kernel composition

deterministic tie rule. Lean proves the excess by a symbolic count argument and transfers it to expected pseudo-regret under the product law. The bound above, with its larger coefficient, is proved for the unchanged policy. The counterexample and the proved bound are two separate certificates. The refutation concerns the stated coefficient under the specified policy, tie rule, and moment model. The source's logarithmic rate and its other estimators are not affected, and no claim is made that the proved coefficient is the smallest possible.

Estimator-level extension. The clipped estimator of Section 5.4 replaces hard truncation by clipping and adds a pathwise corruption budget C on the consumed prefix. Its confidence radius enlarges by C/N at an outcome-dependent count $0 < N \le t$; the proof combines the clean

moment analysis, which calls the two lemmas of this development, with the perturbation lemma. No corruption-robust policy or regret theorem is claimed.

D.2 Lipschitz bandits: hierarchical optimistic optimization

Problem and algorithm. Algorithm 1 and Theorem 6 of Bubeck et al. (2011). A binary tree of cells subdivides the arm space into regions of geometrically shrinking size: at depth h a cell has dissimilarity diameter at most $\nu_1\rho^h$ and contains a ball of radius $\nu_2\rho^h$ around its representative, and these balls are pairwise disjoint across all distinct cells at the same depth, with $0 < \rho < 1$. The dissimilarity need not be a metric. The mean f takes values in $[0, 1]$ and satisfies the weak Lipschitz condition $f^* - f(y) \le f^* - f(x) + \max\{f^* - f(x), \ell(x, y)\}$, reward laws are supported on $[0, 1]$, and $f^* = \sup_{x\in\mathcal{X}} f(x)$ is a supremum: no compactness and no attained optimum is assumed. The algorithm pulls the representative arm of one tree node per round, chosen by optimistic indices.

Reusable mathematics. Adaptive conditional concentration from the actual trajectory; the trajectory construction; Mathlib's countable measurability and Markov-kernel interfaces for the node kernel.

Setting-specific obligations. A bound on visits to poor nodes; packing growth from the near-optimality dimension; a partition of the regret into three disjoint tree classes; optimization of an integer depth.

Guarantee. Let d_0 be the near-optimality dimension at parameter $4\nu_1/\nu_2$. For every $d > d_0$ there is $\gamma > 0$ such that for all $N \ge 1$,

$$\mathbb{E}\Big[\sum_{n<N}(f^* - Y_n)\Big] = \mathbb{E}\Big[\sum_{n<N}(f^* - f(X_n))\Big] \le \gamma N^{(d+1)/(d+2)}\big(\log\max\{N, 2\}\big)^{1/(d+2)},$$

where X_n is the representative pulled at round n; γ is chosen before the horizon quantifier and may depend on d and the geometry but not on N. The equality holds because each reward is drawn from the law of the pulled representative given the past, so its conditional mean is $f(X_n)$, and rewards are bounded.

Source differences. The adapter of Section 5.2 removes a premise of our own first interface. The covering recursion of the source's equation (2) is implemented with the right child one level below its parent, and both children of the root are initialized to infinite indices; the printed and the implemented forms are recorded side by side. The logarithm $\log\max\{N, 2\}$ replaces $\log N$ in the confidence term and in the bound, which agrees with the printed forms for $N \ge 2$ and corrects the logarithmic term at the smallest horizon. Fixed representatives and deterministic left ties instantiate choices the source permits.

Instance. Infinitely many binary-sequence arms with distinct means and non-Dirac rewards; a proved bound $d_0 \le 2$ and the choice $d = 3$ give the exponent $4/5$, without a claim that the dimension or the exponent is sharp.

D.3 Combinatorial bandits: CUCB with triggered arms

Problem and algorithm. In a combinatorial bandit the learner chooses a set of base arms in each round and observes the outcomes of the arms that the chosen set triggers, and because the best set may be hard to compute it may call an approximation oracle. Algorithm 1 and Theorems 1–2 of Chen et al. (2016): actions are subsets of arms with nonempty trigger sets, outcomes lie in $[0, 1]$, and the learner calls an α-approximation oracle that succeeds with probability at least β using fresh randomness after the estimates are formed. The regret is $R(H) = H\alpha\beta\,\mathrm{OPT} - \mathbb{E}\sum_{t\le H} Y_t$, which can be negative; with $d(a) = \alpha\,\mathrm{OPT} - r(a)$ it equals $\mathbb{E}\sum_t d(A_t) - H\alpha(1-\beta)\mathrm{OPT}$, and dropping the oracle-failure credit would change the theorem. The formal statement assumes that the smoothness modulus attains every positive gap up to the largest one, so that it admits an inverse on the positive-gap interval; strict monotonicity alone does not provide this, and the hypothesis is recorded in Table 2.

Reusable mathematics. A conditional exponential-moment bound on the actual action–feedback mixture, introduced during the HOO development; layer-cake integration; a uniform-measure helper hosted in the Thompson module, whose reuse transfers no Thompson-sampling regret theory.

Setting-specific obligations. The normalized-counter charge across the deterministic and random-trigger branches; control of insufficiently observed rounds.

Guarantee. The refined bound keeps each arm's lower-gap threshold term, its layer-cake integral up to the upper gap, and a summable tail residual; a common cutoff gives both polynomial-smoothness bounds for $f(u) = \gamma u^{\omega}$, $0 < \omega \leq 1$, with separate deterministic and random-trigger constants. The exact statements are in the artifact.

Source differences. The source rescales its sampling threshold by trigger probabilities, but the two analysis branches carry different coefficients, so the rescaling is not an identity across branches. The formal analysis charges a round to an arm that minimizes its counter normalized by its own threshold coefficient; the action rule is unchanged. This adapts one analysis step; it does not refute the source's bound.

Instance. Sixty-four equiprobable atoms, three Bernoulli arms with means $1/4, 1/2, 3/4$, overlapping actions $\{0, 1\}$ and $\{1, 2\}$, a fair coin governing observation of the unselected arm, product rewards, and an exact oracle breaking the initial tie toward the inferior action. The instance checks that triggering and reward are distinct objects and that the initial tie produces the computable regret $R(1) = 1/4$.

D.4 Causal bandits with known parent laws

Problem and algorithm. Algorithm 2 and Theorem 3 of Lattimore et al. (2016), with Proposition 4 for uniform allocation and the supplement's parallel-graph bridge. A finite ordered DAG with a binary reward; an intervention replaces the conditional tables of selected variables and never touches the reward node; the learner knows each action's reward-parent law P_a but not the reward table. For an allocation η with mixture $Q = \sum_a \eta_a P_a$, the design must cover every support: at every point where some P_a is positive, Q is positive. This is a hypothesis of the analysis, not a convention about division. Let $c = \max_a \sum_z P_a(z)^2/Q(z)$, $L = \log(2TK)$, and $B = \sqrt{cT/L}$. Each round draws an action from η and then an assignment from that action's intervened joint law; the estimate of action a averages $Y\,(P_a/Q)\,\mathbf{1}\{P_a/Q \leq B\}$, and the recommendation is the least empirical maximizer in a fixed order.

Reusable mathematics. The centering and independent-sum lemmas of Section 5.1 and Figure 3, the fixed-tilt contract they instantiate, and the finite argmax interface.

Setting-specific obligations. The intervened joint law of one sample, constructed in topological order; the mean and second moment of the truncated importance statistic; the simultaneous confidence event over actions with failure probability at most $1/T$; the recommendation step. For the designs: uniform allocation covers all supports with $c \leq K$; an attained optimal allocation exists, proved by a compactness argument on a covered sublevel set of the cost, which the library establishes rather than assuming that an optimal design is given; and for $N \geq 2$ independent binary roots a normalized parallel design has cost at most twice the constructed rarity parameter, deterministic roots included. The learner's threshold always uses the design's true cost; only the right-hand side of the bound is relaxed.

Guarantee. $\mathbb{E}R_T \leq (2\sqrt{2}+7)\sqrt{cL/T} + 1/T$ and $\mathbb{E}R_T \leq 1$, where R_T is the simple regret of the recommendation.

Source differences. The displayed proof's final constant $\sqrt{2}+4$ counts one confidence radius plus the bias, while the chain $\mu_* - \mu_I \leq \varepsilon_* + \varepsilon_I + \beta_*$ contains two radii; the written inequalities give $2\sqrt{2}+7$, and no claim is made that smaller constants are impossible. The final probability display bounds the good event where the bad one is meant; the proved statement is $\Pr(\text{bad}) \leq 1/T$. The

printed parallel design has total mass $3/2$; mass $1 - D$ normalizes it, and the counting bound requires the strict rarity condition. The amended guarantees retain the rate order in the stated model with these changed constants and conditions.

Instance. The DAG $X \to W$, $X \to Y$, $W \to Y$ with $X \sim \text{Bernoulli}(1/2)$, $\Pr(W{=}1 \mid X) \in \{1/4, 3/4\}$, and $\Pr(Y{=}1 \mid X, W) = (1 + X + 2W)/5$ has intervention means $1/2$, $3/10$, and $7/10$ for observation, $\text{do}(W{=}0)$, and $\text{do}(W{=}1)$, while the conditional mean given $W = 1$ is $3/4$. The two values for "$W = 1$" differ because observing $W = 1$ changes the law of X, whereas the intervention does not, and the two computations use two different laws in the library. Observation-only allocation covers all parent states with cost $8/3$; at budget one the tuned threshold lies in $(1, 4/3)$ and the exact expected simple regret is $1/5$. Fair and deterministic two-root parallel instances have design costs 2 and 4.

D.5 Multi-player bandits: Musical Chairs

Problem and algorithm. Algorithms 1–2 and Theorem 1 of Rosenski et al. (2016). There are n players and K arms with $0 < n < K$; players are synchronized, share stationary arm means on $[0, 1]$, sense their own collisions, do not know n, and know a positive lower bound ϵ on the gap between the nth and $(n{+}1)$th largest means. Each player estimates the population from collision frequencies, ranks arms from collision-free rewards, and samples retained candidates until a collision-free pull fixes an arm.

Reusable mathematics. None from the initial library (Appendix G); the development is self-contained.

Setting-specific obligations. Population estimation from collisions; ranking; coordination; and the distinction between the conditional and the unconditional regret quantities below.

Guarantee. With exploration length $S = \lceil \max\{16K\epsilon^{-2}\log(4K^2/\delta),\ 50K^2\log(4K/\delta)\} \rceil$ for a fixed $\delta \in (0, 1)$, the expected aggregate pseudo-regret at every horizon H is at most $\min\{nH,\ n\min\{H, S\} + 8n^2 + \delta nH\}$, and the visible reward process satisfies an unconditional bound with the coarser charge nS. For fixed δ the term δnH is linear in H; a horizon-dependent choice of δ, which retunes S, is not formalized. Conditional pseudo-regret and unconditional realized expectation are kept separate because conditioning on the reward-dependent exploration event changes the reward means, an instance of the distinction between criteria in Section 4.

Source differences. The source's coordination theorem takes a correct common candidate set as an input; the library supplies the statistical producer that yields it, closing the chain from learning to coordination. The population-estimation logarithm is enlarged to pay for an all-player union bound, a disclosed change to the exploration budget. Neither is a refutation of the source's theorem.

Instance. Two players and three Bernoulli arms with means $3/4, 1/2, 1/4$ under the repaired exploration length; a specified event combining successful exploration and transient coordination has probability at least $3/64$, with mean coordination charge $5/4$. The instance checks that the interfaces are non-degenerate.

E Routes in progress

The three routes below are the named leaves of the Frontier's formalization queue (Section 3). Each is described by what is proved, what the source's argument needs next, and what remains open.

Delayed best-of-both-worlds. The source (Schlisselberg et al., 2025) analyzes one algorithm, with one tuning, in both stochastic and adversarial environments; the library binds that algorithm, its initialization, its monitoring mechanism, and its tuning across both regimes, with the source's delay model and comparator. The accounting layer defines, at each decision time t and over the earlier rounds $s < t$, the set of rounds whose feedback has arrived, $s + d_s < t$, and its complement, $s + d_s \geq t$, proves that the two sets partition the past and that their sizes add to t, and relates the

source's one-based end-of-round count of missing feedback to this zero-based action-time count through an explicit index shift. The algorithmic layer formalizes how the algorithm processes delayed feedback in the source's order, maintains its active arms, eliminates arms, and opens a new phase for each newly eliminated arm i with the source's parameters

$$p_i^1 = \frac{1}{2K} + \frac{n_i(S)}{2T}, \qquad \widetilde{\Delta}_i = 8w_i(S), \qquad N_i^1 = \frac{1280}{p_i^1 \widetilde{\Delta}_i^2},$$

where S is the processed state at the elimination time, $n_i(S)$ the pull count of arm i in it, and $w_i(S)$ its confidence width; positivity of the denominators is proved for $T > 1$. The probabilistic layer assembles the source's union bound over good events. The confidence width of Lemmas D.10–D.12 decreases with the count while the appendix uses it in the increasing direction; the library proves the consequence conditionally on the used direction and records the discrepancy, which remains open. Still to be established are the evolution of the confidence bounds (Lemma D.13), the simultaneous probability statement (D.4), the switching argument between regimes, and the two regret theorems.

Stochastic-gradient bandit: the latent-stream bridge. Theorem 1 and Corollary 1 of Baudry et al. (2025) are complete in the two-arm setting with independent rewards under the source's remaining hypotheses, which are listed in the artifact. For Theorem 2 the formalization follows the phase argument of the source's appendix one interface at a time (Table 5). Its central device is a latent process in which each arm has an independent stream of rewards. The library proves that the rewards the algorithm observes have the same law as the corresponding entries of these streams on every finite prefix, and extends the identity to the whole trajectory law by a uniqueness argument over the generating prefix events.

Stochastic-gradient bandit: missing pulls. Because the algorithm may stop pulling an arm, the nth observation of that arm may never occur. The library encodes missing observations explicitly, transports the source's phase event to the generated process, and splits its probability into the case in which every needed pull occurs and the case in which some pull is missing. The missing case yields a finite-horizon charge to the expected regret; it does not yield a positive probability bound, and no claim is made that the observed rewards are independent conditionally on all pulls occurring.

Stochastic-gradient bandit: the next open leaf. The source's trigger event for the phase remains open. Separately, a step in the appendix proof of the source's Theorem 4 alternates between conditioning on $q \geq c$ and on $q < c$, and the displayed probability bound does not supply the lower bound the preceding step needs; this is recorded against the source and neither refuted nor repaired.

Table 5: Route to Theorem 2 of the stochastic-gradient source: mathematical step, evidence in Lean, and what remains before the next step.

Mathematical step	Evidence	Remaining requirement
observed rewards and latent streams agree on every finite prefix	proved	extension to whole trajectories
agreement of whole trajectory laws	proved by uniqueness over prefix events	law of rewards at selection times
law of the rewards at the times an arm is selected, with missing pulls encoded	proved	transport of the phase event
the source's phase event on the generated process	proved	split of its probability
split into "every pull occurs" and "some pull missing"	proved	link from a missing pull to a low final count
a missing pull implies a low final count	proved	regret charge for the missing case
finite-horizon regret charge for the missing case	proved	the source's trigger event for the phase: open

Succinct lower bound. The source (Zeng & Honorio, 2025) defines a set of unit atoms, a supremum Q of their correlations, and a dual quantity $R(X)$ as a supremum over $Q(Y) \leq 1$. Lean keeps these

suprema in the source's real-valued form and proves Definitions 3.1–3.3 and Lemmas 3.1–3.4. It then proves that if X has a nonzero component orthogonal to every atom, the set over which $R(X)$ is a supremum is unbounded, so R cannot be a finite real-valued function on the whole space when the atoms do not span it. Adding a spanning premise, restricting the domain to the span, and moving to an extended-real codomain are three different amendments with different downstream obligations; none has been adopted. Independently, the proof of Theorem 3.8 uses two sample-size regimes that its statement omits. Theorem 3.8 remains open.

F Development and review records

Every source-facing development keeps three kinds of record. The mathematical claim consists of the source statement with its version and anchor, the formal statement, and the assumptions it may use, summarized in a short signature that names the algorithm, the feedback model, the performance criterion, and the order of quantifiers. The proof obligations consist of the dependency graph of intermediate results and the obligations still open. The publication evidence consists of the verified declarations and the reader page. The claim is fixed and the proof strategy is free: contributors may change the decomposition, generalize a shared lemma, or retract an intermediate definition, while changes to assumptions or conclusions must be disclosed and reviewed.

Because Lean checks a proposition and not its correspondence to a cited theorem, each development also carries a source comparison. One party writes the formal statement. A second party receives the statement, the definitions it depends on, and neutral notation, but not the source, and writes out the mathematical claim with all of its assumptions. A third party compares this reconstruction with the source in seven respects: the objects and spaces, the order of quantifiers, the assumptions, the conclusion and its performance measure, the constants, the probability and feedback model, and the part left unproved. Each difference is classified as implicit in the source, mathematically necessary, an interface limitation, a generalization, or unresolved. The HOO premise of Section 5.2 was found in this way: the reconstruction of the first statement reported a measurability assumption that the comparison could not find in the source. In this project the parties were isolated AI instances under the authors' supervision. The records of the five focal developments, and a pre-audit of eighteen contracts derived from three recent papers, are provided with the artifact together with the reviewed file versions. They describe review agreement and are not accuracy against an independent standard.

Before a development becomes public, mechanical checks build the library root and test modules under the pinned toolchain, verify that every theorem is reachable from the public root and exercised by a typed instantiation, scan for unproved placeholders, audit that public declarations use only Lean's standard axioms, recompute the digests of reviewed modules and stop the site build when they differ from the recorded ones, and regenerate every index and status from recorded data. The authors then judge whether the reader page states the claim the formal development supports.

G Reference counts per development

For each of the five focal developments we extract, at this release and from elaborated proof terms, the direct reference pairs from its modules to library declarations outside them, and we attribute every referenced file by Git object comparison to the library as it stood before the focal developments began (Table 6). The figures locate where existing foundations were used. They count generated declarations, exclude Mathlib, and say nothing about effort or about how much of a proof a reference carries. The Musical Chairs development's 902 internal reference pairs document integration within the development rather than reuse.

Table 6: Direct reference pairs from each development's modules to library declarations outside them, at this release. "Unchanged" files are identical to their version before the focal developments began; pairs to files added during these developments are listed separately. Generated declarations are included; Mathlib references are excluded.

Development	Modules	Distinct referenced declarations	Pairs	To unchanged files	To files added during the developments
Heavy-tailed robust UCB (with clipping extension)	29	63	461	419 (20 files)	42, one arm-stream policy file
HOO	29	21	36	28 (5 files)	8, four files
CUCB	38	15	38	29 (4 files)	9, incl. the conditional moment interface
Causal bandits	20	8	20	18 (3 files)	2, the heavy-tailed lemmas
Musical Chairs	12	0	0	0	0 (902 pairs inside the development)

H FORMALIZATION EFFORTS COMPARED BY ARTIFACT AND EVIDENCE

Table 7 compares the efforts of Section 2 by what they produce and what they report, as described in their own papers. Cells marked "not reported" record the absence of a description in the paper, not the absence of the capability.

Table 7: Formalization efforts by artifact, reader or reuse support, and reported evidence, summarized from each paper's own description.

Work	Mathematical artifact	Reader or reuse support	Evidence reported
AI4SLT (Zhang et al., 2026b)	statistical learning theory library, sorry-free	declaration-to-textbook table; dependency figure	scale; gaps that textbooks skip
FoML (Sonoda et al., 2025)	Rademacher and Dudley bounds on Mathlib	coverage tables by system	measurability conditions made explicit
RL theory (Zhang, 2025)	convergence of TD and Q-learning under Markov noise	not reported	proof artifact; a limitation of the rate parameter
LeanMarathon (Zhang et al., 2026c)	formalizations of two research papers	evolving blueprint as system of record	lemma counts; cost; baseline comparison
LeanFlow (Milikic et al., 2026)	formalizations of two papers; benchmark slice	source-backed skeleton	workflow ablation; expert inspection
Trellis (Pegden, 2026)	formalizations of two Ramsey-theory papers	human-ratified semantic closure	line counts; verifier pass rates
Automatic Textbook Formalization (Gloeckle et al., 2026)	one textbook, generated	not reported	scale and cost; manual spot checks
M2F (Wang et al., 2026b)	literature-scale formalization	not reported	compile rates; expert audit as filter
LCS-Bench (Feng et al., 2026)	one logic textbook	loophole catalogue	equivalence checker; model pass rates
BanditRLlib (this work)	textbook backbone, five bandit developments, finite-horizon RL with UCB-VI	natural-language/Lean workspace; Lean Graph; mechanism hypergraph; BanditRLwiki and Frontier	compiled reuse; recorded source differences; author-supervised AI review