EVOLVE FRAMEWORK

Anonymous authors
Paper under double-blind review

1 PRELIMINARIES

A discovery task d defines the scientific objects that may be proposed, the evidence required to evaluate them, and the resources available for search. Each completed allocation runs one search branch and returns a candidate x with an immutable evidence packet η. An independent verifier decides whether the candidate is admissible and assigns a bounded verified score $R_d(x, \eta) \in [L, U]$. Scores are oriented so that larger values are better. When evaluation is noisy, R_d is a prespecified lower confidence value rather than a single measurement.

Let C_i be the cost of the ith completed branch. Under total budget B, the final completed branch is

$$\tau_B = \max \left\{ t \;\middle|\; \sum_{i=1}^{t} C_i \le B \right\}. \tag{1}$$

The verified record after t branches is

$$M_t = \max \left(M_0, \max_{\substack{1 \le i \le t \\ x_i \text{ is admitted}}} R_d(x_i, \eta_i) \right). \tag{2}$$

EVOLVE seeks a policy Π that maximizes the expected final record

$$J_B(\Pi) = \mathbb{E}_{\Pi} \left[M_{\tau_B} \right]. \tag{3}$$

Admissibility, scientific diversity, and resource limits guide which branches may be run. They are not added to the verified score. This keeps the search focused on the best confirmed scientific result.

2 FRAMEWORK

We call the framework EVOLVE for Evidence-Verified Option Learning and Value Exploration. EVOLVE organizes discovery as repeated branching from a verified scientific archive. A scheduler chooses where to begin, which agent role to use, which multi-step search tactic to run, which harness version should support it, and how much budget it receives. An independent verifier then returns evidence that updates the archive, the scheduler, the option memory, and the selected agent policy.

The default system uses one frozen language model backbone with three separate LoRA adapters. The adapters implement the Scout, Mechanist, and Challenger roles. Each role has its own optimizer state, random stream, and private working memory. This gives EVOLVE three evolving policies while keeping the shared foundation model fixed.

2.1 EVOLVE AT A GLANCE

Search is divided into epochs. An epoch is the interval between two synchronization barriers. The following sequence gives the complete operating loop.

1. Freeze the current role adapters, archive map, harness versions, verifier version, and record threshold.
2. Select promising and under-tested scientific states from the archive.
3. Form candidate allocations by combining a state, role, option, harness version, horizon, and cost.

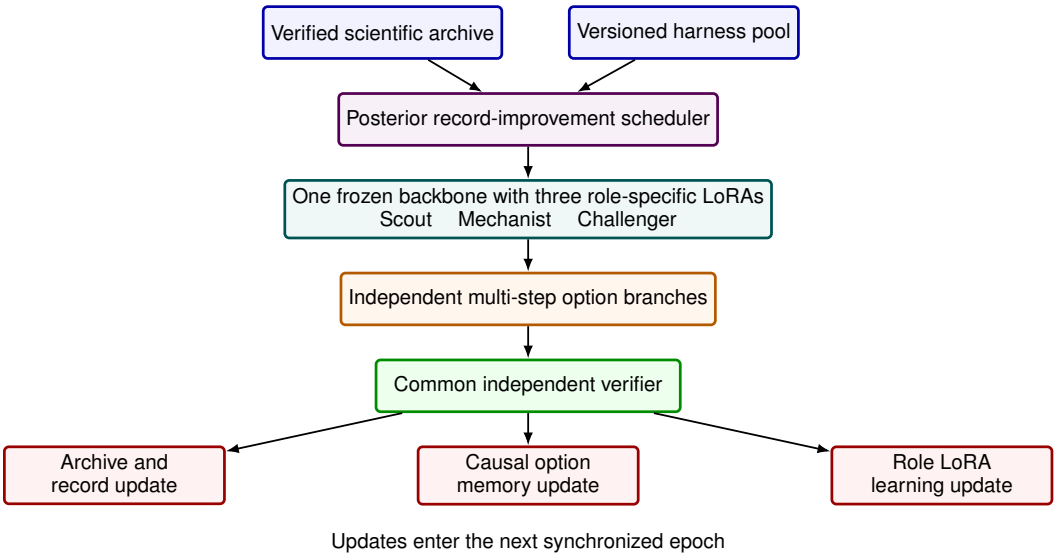

Figure 1 The main EVOLVE cycle. Archive states and harness versions enter the scheduler. A selected role runs an option branch under one frozen backbone. Verified evidence then updates the archive and record, causal option memory, and the corresponding role LoRA for the next epoch.

4. Reserve small budget shares for randomized audits, archive coverage, all three roles, and new harness trials. Allocate the remaining budget by expected record improvement.
5. Run independent production branches and matched audit pairs.
6. Verify every completed branch and optionally send a suitable provisional candidate through bounded refinement.
7. At the synchronization barrier, update the record, archive, scheduler, causal option memory, and the LoRA adapter of each role from its own verified data.

The same evidence is used in different ways. Production branches estimate where the next record is likely to come from. Randomized audit pairs estimate whether an option or harness caused an improvement relative to a matched alternative. Homogeneous groups of completed branches provide the rewards for test-time policy learning.

2.2 Scientific States and Option Branches

The archive groups candidates by scientific behavior rather than by wording or surface form. A versioned descriptor $\phi_v(x, \eta)$ summarizes measured responses, mechanistic structure, and active constraints. A mapping q_v assigns the descriptor to a scientific state cell

$$c = q_v\big(\phi_v(x, \eta)\big). \tag{4}$$

Text similarity may help retrieval, while the scientific descriptor determines cell identity. The descriptor version remains fixed during an epoch so that evidence collected in the same epoch is comparable.

A cell keeps several useful candidates instead of retaining just one. These include its strongest confirmed candidate, a candidate with promising descendant value, and a small set of stepping stones. Local competition compares candidates within nearby scientific states. The best confirmed candidate across the whole archive is also retained as the global record holder. A coverage budget gives empty and under-tested cells regular opportunities without changing the scientific score.

This archive follows the local competition idea of quality-diversity search while allowing uncertain descriptors and more than one useful candidate per cell (Mouret and Clune, 2015, Flageat and Cully, 2023).

An option is a reusable multi-step search tactic

$$o = (I_o, \pi_o, \beta_o, H_o). \tag{5}$$

Here I_o describes where the tactic may begin, π_o is its internal proposal policy, β_o is its stopping rule, and H_o is its largest branch budget. Examples include exploring a new representation, modifying an active mechanism, challenging an assumption, or refining a provisional candidate.

This definition adapts the options formalism from temporal abstraction to scientific search (Sutton et al., 1999, Bacon et al., 2017).

Launching an option from a cell creates a local descendant tree. Different trees may later reach the same scientific cell, so the persistent search structure is an archive-linked provenance graph rather than one global tree. The tree records how a result was reached. Allocation decisions are made by the scheduler described below.

For a branch with horizon H, let Z_H be the best verified score found by that branch or any of its descendants. The branch gain is

$$G_H = \left[Z_H - M_e^{\text{start}}\right]_+, \tag{6}$$

where M_e^{start} is the record frozen at the beginning of the epoch. This gives useful credit to a tactic that reaches a productive region even when its first proposal does not improve the record.

2.3 Three Isolated Agent Roles

EVOLVE uses one frozen backbone and three role-specific LoRA adapters.

Scout The Scout searches remote and underexplored cells. It proposes new option families, representations, and directions that may broaden the scientific search space.

Mechanist The Mechanist works from current constraints, mechanisms, and invariants. It develops targeted options whose value may appear over several descendant steps.

Challenger The Challenger searches for counterexamples, boundary cases, and minimal changes that test a current proposal. Its candidates are evaluated by the same independent verifier used for the other roles.

Each role keeps a private working memory and retrieval view. Raw transcripts remain private to the role that produced them. After verification, compact evidence packets become available to the other roles at the next synchronization barrier. Gradients update the corresponding role adapter and are not merged across roles. The result is a state-isolated ensemble rather than a claim that the three roles are statistically independent.

2.4 Adaptive Allocation and Harness Selection

A harness is the operating setup around an agent. It specifies the archive view, prompt and context assembly, role routing, branch breadth and depth, resource envelope, and verification schedule. EVOLVE does not rely on one harness being best in every context. Instead, the scheduler treats the harness as part of the allocation.

This choice is motivated by evidence that discovery harnesses can rank differently across models and tasks (Gupta et al., 2026).

Let h_v denote a harness and its version. A harness version remains fixed inside a branch. Any change to its context, routing, search schedule, memory access, or verification timing creates a new version that enters through a small budget-matched trial allocation. This makes the harness adaptive across branches and stable within each branch.

An allocation arm is

$$r = (c, a, o, h_v, H, C), \tag{7}$$

where c is the starting cell, a is the role, o is the option, h_v is the harness version, H is the branch horizon, and C is the total cost. Let Z_r be the verified descendant maximum predicted for arm r.

Given remaining budget b_t, the scheduler selects a portfolio S_t by

$$S_t \in \arg \max_{S \mid \sum_{r \in S} C_r \leq b_t} \mathbb{E} \left[\left[\max_{r \in S} Z_r - M_t \right]_+ \middle| \mathcal{D}_t \right]. \tag{8}$$

This is posterior expected record improvement. It favors portfolios that are likely to beat the current record. The joint prediction also reduces the value of selecting several branches that are individually promising but likely to return the same result.

Before this performance-based allocation, fixed budget shares support randomized audits, empty or under-tested cells, all three roles, and provisional harness versions. The remaining budget follows Eq. (8). Production outcomes update the tail posterior used by the scheduler. Admission probability and the score distribution of admitted candidates are modeled separately because both affect the chance of a new record.

2.5 Verification, Feedback, and Causal Memory

All harnesses use the same external verifier. The verifier checks admissibility, evaluates the primary scientific measure, applies controlled perturbations when appropriate, and repeats measurements when outcomes are stochastic. It returns a compact certificate containing the admission decision, verified score, scientific descriptor, useful diagnostics, and branch lineage. Information reserved for blinded evaluation remains hidden from the agents.

EVOLVE uses three forms of memory. The archive stores verified scientific states. Each role keeps private working memory for its current reasoning. A causal option registry stores reusable evidence about options and harnesses. This registry contains structured records rather than unrestricted summaries. A record identifies the intervention, matched alternative, starting context, assignment probability, cost, horizon, verified outcome, uncertainty, and verifier version.

Production data alone may overrepresent options that the scheduler already prefers. A fixed share of the budget is therefore used for randomized audit pairs. Both branches start from the same state and frozen role checkpoint. They receive the same horizon and cost, while one branch receives the option and the other receives a matched continuation o_0. Their outputs remain separate until both branches close. The local branch effect is

$$\tau_H(c, a, h_v, o) = \mathbb{E} \left[G_H(o) - G_H(o_0) \mid c, a, h_v \right]. \tag{9}$$

Here causal refers to the effect of invoking a search tactic relative to its matched continuation. It does not refer to causal laws inside the scientific domain. Known assignment probabilities and branch isolation allow the registry to estimate this effect within the audited context. Harnesses can be compared in the same way by changing the harness version while holding the state, role, option, horizon, cost, and verifier fixed.

An option enters long-term causal memory after repeated audit evidence supports a positive contextual effect. Retrieval is conditioned on the current state, role, harness, and horizon. A synthesized natural-language strategy is evaluated as a new option before it receives the status of an established memory. A small no-memory audit arm continues to measure whether retrieval remains useful over time.

Feedback follows three clear paths. Verified outcomes update the archive and scheduler. Matched audit outcomes update causal memory. Homogeneous groups of verified branches update the role adapters. Keeping these paths separate makes it clear whether an observation supports search allocation, causal attribution, or policy learning.

2.6 Test-Time Policy Learning

This is the reinforcement-learning component of EVOLVE. Learning occurs after a group of branches has closed. Every branch in a learning group shares the same scientific cell, role, option, harness version, horizon, cost range, policy snapshot, and frozen record threshold. This makes the reward comparison meaningful.

Suppose K branches are sampled from one frozen role policy and their verified gains are sorted as $G_{(1,K)} \leq \cdots \leq G_{(K,K)}$. EVOLVE optimizes the mean of the largest m gains

$$J_{K,m}(\theta) = \mathbb{E}\left[\frac{1}{m}\sum_{j=K-m+1}^{K} G_{(j,K)}\right]. \tag{10}$$

Pure Max at K is recovered when $m = 1$. A slightly larger m keeps learning focused on the upper tail while reducing sensitivity to a single extreme outcome.

A larger on-policy batch with $N > K$ branches can be reused across subsets of size K. OrderGrad provides the corresponding finite-sample policy-gradient estimator under its sampling assumptions (Parmas et al., 2026). MaxPO provides an exactly centered alternative for the pure maximum objective (Takashiro et al., 2026). The log probability of a branch is the sum of the role policy log probabilities for its internal decisions. After a group closes, its gradient updates the LoRA adapter of that role. The shared backbone and the other two role adapters remain unchanged.

The scheduler and the role policies learn different things. The scheduler learns which allocation is likely to produce the next record. Role learning improves how a selected option is carried out. Both use verified descendant maxima, which keeps search and training aligned with Eq. (3).

2.7 Bounded Refinement

A provisional candidate with a clear and actionable diagnosis may enter a bounded refinement stage. It receives one to three attempts, a fixed cost limit, a maximum depth of two, and no second entry after leaving the stage. The Challenger proposes the smallest change that may address the diagnosis. Each revision is treated as a new candidate and returns to the blinded verifier with a new evidence packet.

Refinement is also represented as an option. Eligible provisional candidates in the audit channel are randomly assigned either to refinement or to an equal-cost fresh continuation from the same starting state. This comparison estimates when refinement is useful. A refined candidate updates the archive, causal memory, or role learning data after independent confirmation.

When the total budget is exhausted, EVOLVE returns the verified record holder together with its evidence packet and complete lineage.

2.8 Detailed Framework Views

The following diagrams expand the main EVOLVE cycle into six complementary views. Each view follows the same evidence flow while isolating one part of the framework that benefits from a closer explanation.

End-to-End Execution The complete pipeline begins from the verified scientific archive, promoted causal option records, and versioned harness pool. The scheduler forms allocation arms, protects support for audits and coverage, and dispatches production branches and randomized audit pairs through isolated role policies. Independent verification routes evidence to the archive, scheduler, causal memory, and role learning before accepted updates cross the synchronization barrier together.

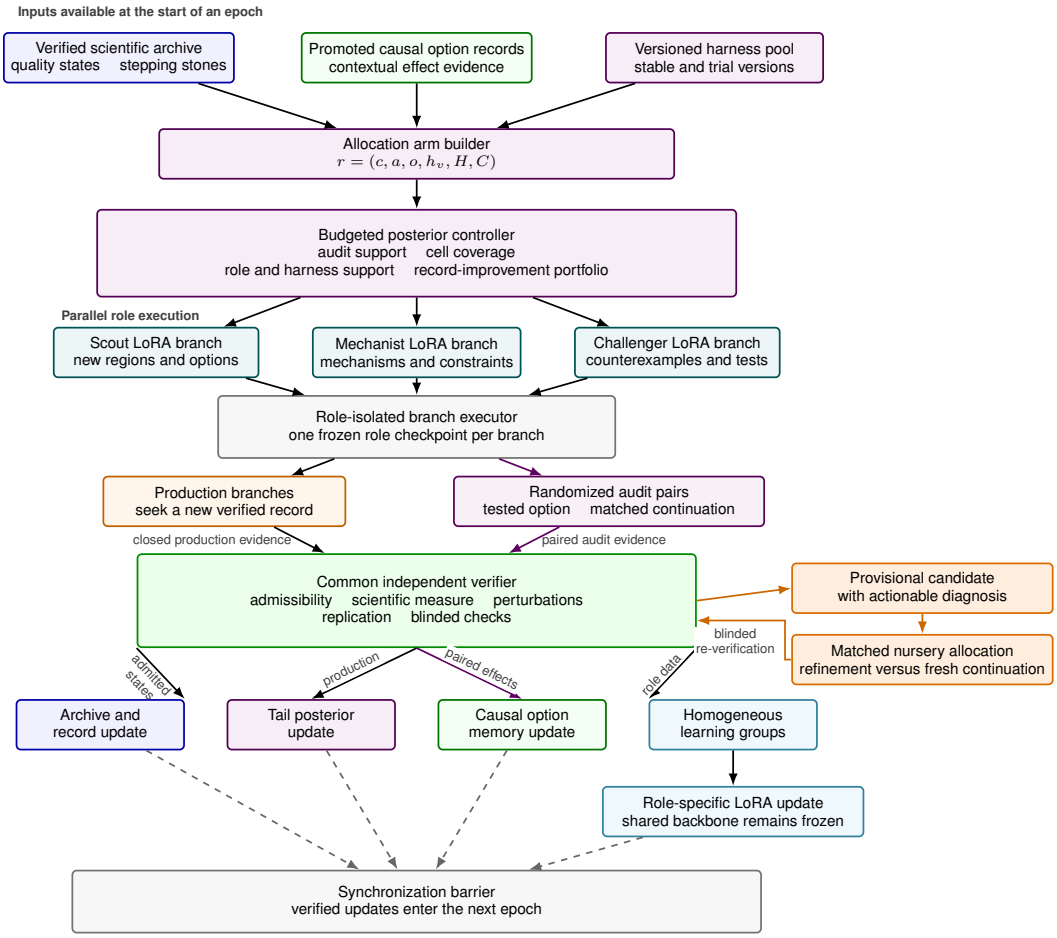

Figure 2 Detailed EVOLVE execution across allocation, isolated role branches, independent verification, bounded refinement, and synchronized updates.

Archive and Provenance The scientific archive is persistent, while every launched option creates a local tree of descendants. The verifier and frozen descriptor map decide whether a candidate is admitted and which scientific cell receives it. A separate candidate-level provenance graph preserves route identity and complete lineage even when different branches reach the same cell.

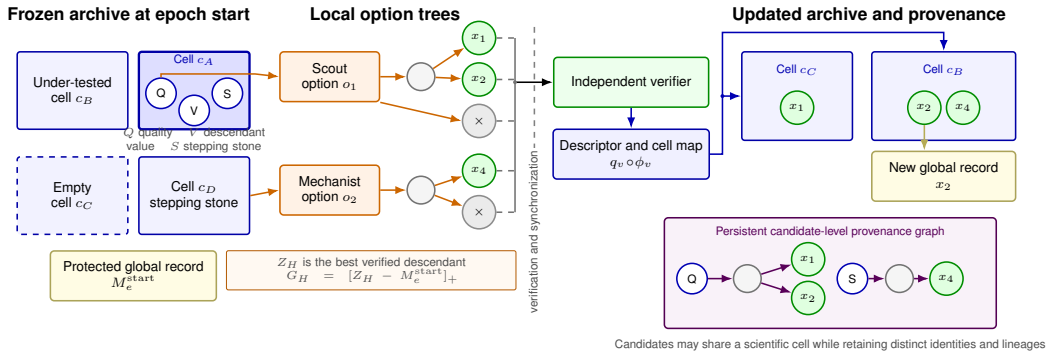

Figure 3 Archive cells, local option trees, descriptor-based placement, and the persistent candidate provenance graph.

Agent and Memory Isolation Scout, Mechanist, and Challenger share one read-only backbone and a verified public evidence layer. Each role keeps its own LoRA adapter, optimizer, random stream, working memory, retrieval view, and raw branch transcript. Only compact verified packets cross the synchronization barrier, and each role learns from its own homogeneous verified group.

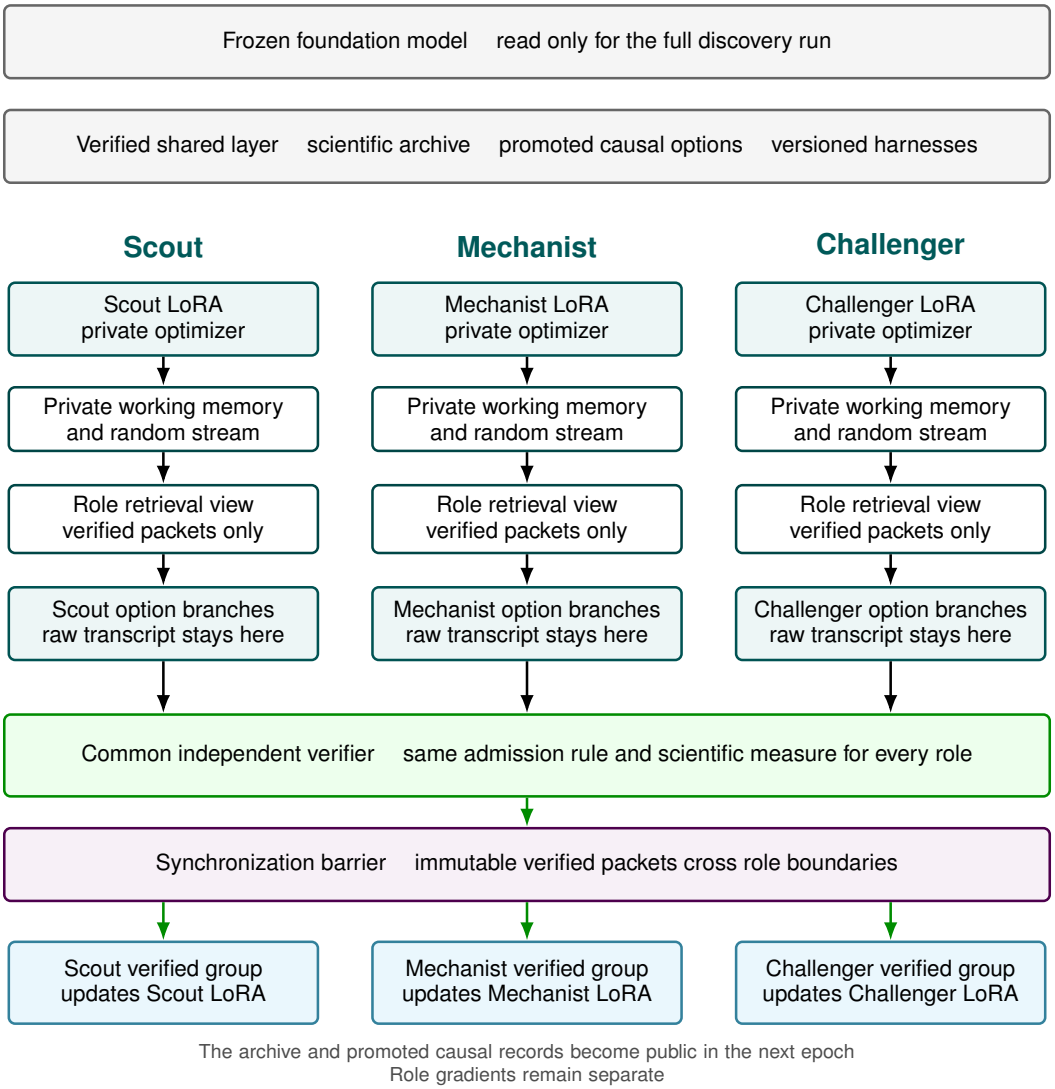

Figure 4 Isolation of role adapters, working memories, retrieval views, branch transcripts, and verified learning updates.

Adaptive Harness Allocation The harness version is part of every allocation arm alongside the state, role, option, horizon, and cost. The scheduler reserves support for audits, scientific coverage, every role, and provisional harness trials before allocating the remaining budget toward expected record improvement. A selected harness remains locked throughout its branch, while later evidence can change which version is chosen for future branches.

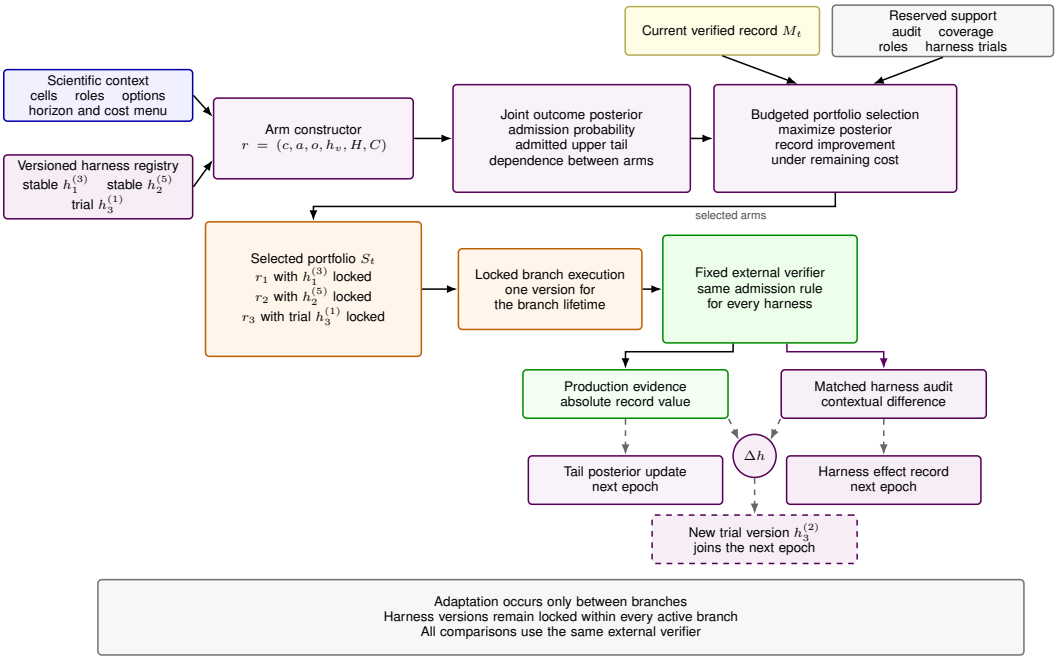

Figure 5 Adaptive allocation over state, role, option, harness version, horizon, and cost with locked branch execution.

Randomized Audits and Causal Memory An option branch and its matched continuation begin from the same frozen state and role checkpoint under the same harness, horizon, cost, and verifier. Their paired difference estimates the local effect of invoking the option rather than the selection preference of the scheduler. Repeated positive audit evidence can promote a context-conditioned record into causal option memory, while uncertain evidence leads to further matched audits.

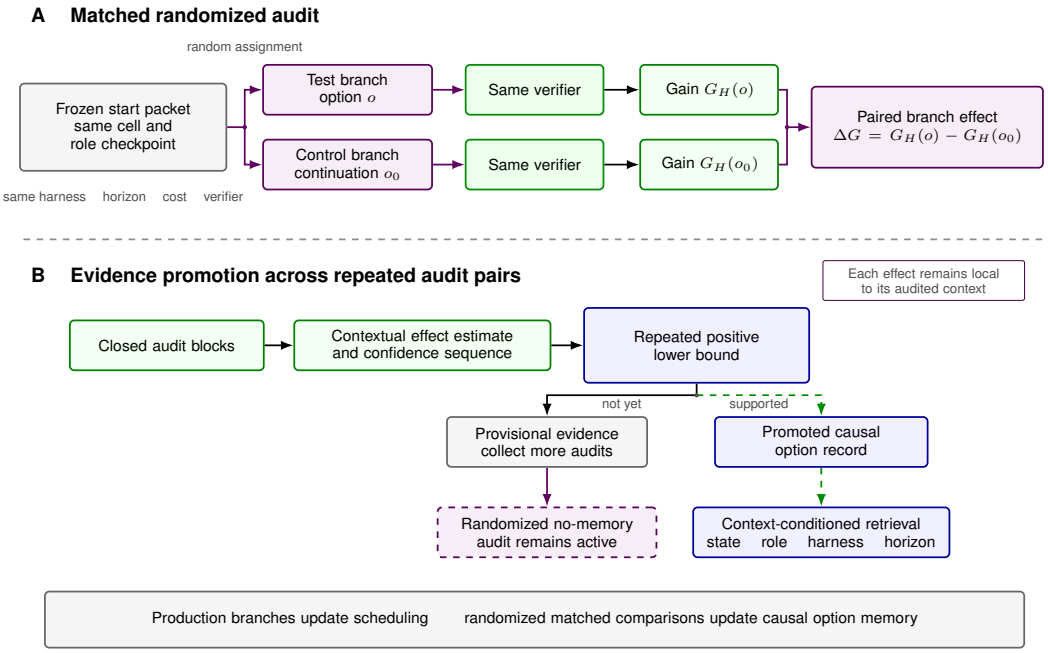

Figure 6 Matched randomized audit pairs and evidence promotion into context-conditioned causal option memory.

Test-Time Learning and Refinement Homogeneous on-policy groups produce an upper-tail update for one selected role while the backbone and the other role adapters remain fixed. Provisional outcomes follow a separate bounded path that permits only a few diagnosis-guided revisions. Every revision returns to independent verification, and only confirmed revisions enter their own refinement learning group.

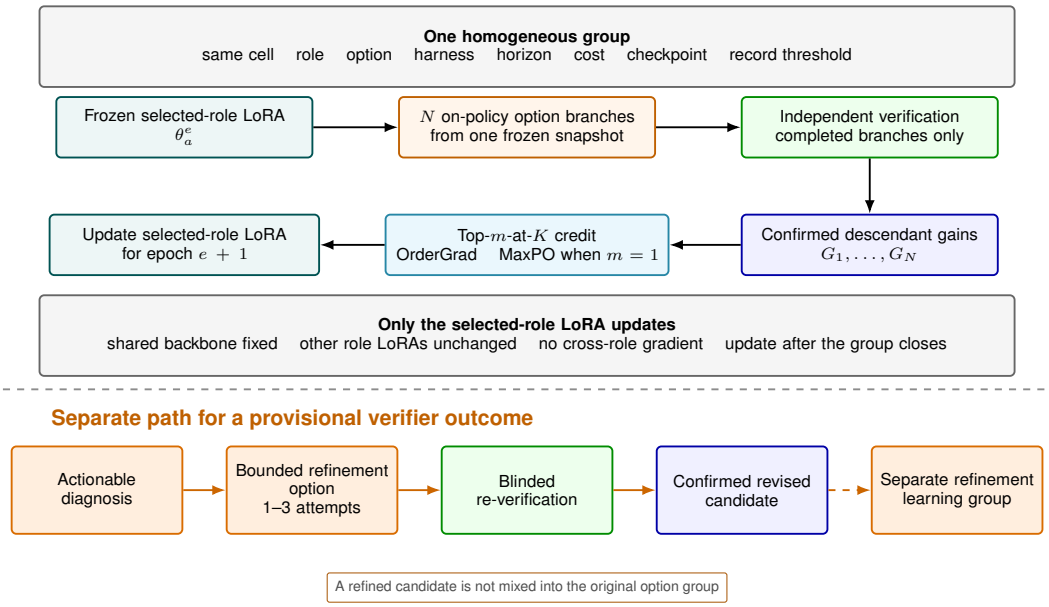

Figure 7 Test-time role learning from homogeneous verified groups and the separate path for bounded refinement.

REFERENCES

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

Manon Flageat and Antoine Cully. Uncertain quality diversity with evaluation methodology and new methods for uncertain domains. *arXiv preprint 2302.00463*, 2023.

Akshat Gupta, Jermaine Lei, Alexander Lu, Gopala Anumanchipalli, and Leshem Choshen. Automated discovery has no universally superior harness. *arXiv preprint 2607.18235*, 2026.

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv preprint 1504.04909*, 2015.

Paavo Parmas, Yongmin Kim, Kohsei Matsutani, Shota Takashiro, Soichiro Nishimori, Takeshi Kojima, Yusuke Iwasawa, and Yutaka Matsuo. OrderGrad and optimizing beyond the mean with order statistic policy gradient estimation. *arXiv preprint 2606.06096*, 2026.

Richard Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs with a framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112, 1999.

Shota Takashiro, Soichiro Nishimori, Paavo Parmas, Yongmin Kim, Kohsei Matsutani, Gouki Minegishi, Yusuke Iwasawa, Takeshi Kojima, and Yutaka Matsuo. On advantage estimates for Max at K policy gradients. *arXiv preprint 2606.06080*, 2026.