ASCT: ATTENTIVE SEARCH OVER COUNTERFACTUAL TREES FOR CREDIT ASSIGNMENT IN AGENTIC REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Terminal utility evaluates a complete agentic workflow, but learning requires credit for the decisions within it. We introduce Attentive Search over Counterfactual Trees (ASCT), a framework that turns training-time multi-step search into local action credit. At actor-visited states, an auxiliary tree evaluates alternative legal actions from the same recoverable prefix. Its action-value table is centered by the frozen actor's probabilities and supplies credit for PPO on actor-sampled trajectories. This protocol connects counterfactual evaluation to policy learning while deploying the actor alone. Uniform, UCT, and cost-aware AgentUCT instantiate the framework. On HotpotQA agentic retrieval-augmented generation, all three improve mean held-out utility over trajectory-return PPO and workflow-adapted VinePPO. Across three seeds, ASCT-AgentUCT reaches 0.6187 utility versus 0.5939 for VinePPO, with gains in answer F1 and execution cost, and uses 50.3% fewer recorded auxiliary Qwen tokens. Transfer and component-description studies examine the learned policies beyond the training setting.

1 INTRODUCTION

A workflow actor, the trainable large language model (LLM) policy in this paper, chooses components, gathers evidence, and decides when to stop. Terminal utility combines answer quality and execution cost, but a successful workflow can still contain a poor local choice. Learning requires credit that connects each decision to its downstream consequences.

VinePPO estimates state values from Monte Carlo continuations for step-wise credit (Kazemnejad et al., 2025); branching learners derive updates from collections of search trajectories. We study an action-resolved interface: compare executable alternatives at states the actor visits, then use those comparisons to train its own sampled decisions.

Recoverable prefixes make this possible. From the same observations, an evaluator can execute alternative legal actions through later workflow decisions to terminal outcomes. Their value depends on these continuations: different retrieved evidence can change whether another retrieval round is useful. Finite action sets permit root coverage, while shared prefixes permit execution reuse within a limited auxiliary budget.

We introduce Attentive Search over Counterfactual Trees (ASCT). To our knowledge, ASCT is the first agentic RL framework to use multi-step counterfactual tree search as an auxiliary legal-action value estimator for actor-centered credit on separately sampled actor trajectories. Search constructs a local Q table; frozen actor probabilities center its values; PPO updates the actor's sampled decisions with their recorded behavior probabilities. Deployment uses the actor alone.

All three evaluators—Uniform, UCT, and AgentUCT—improve mean held-out utility over trajectory-return PPO and workflow-adapted VinePPO. Across three seeds, ASCT-AgentUCT reaches 0.6187 versus 0.5939 for VinePPO, combining higher answer F1 with lower execution cost and using 50.3% fewer recorded auxiliary Qwen tokens. Transfer and description-controlled experiments examine the learned policies beyond the training setting.

ASCT connects: **multi-step action comparison** from the same recoverable state; **separate evaluation allocation and behavior sampling**, linked by actor-centered credit and consistent PPO records;

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.

and **training-time information acquisition amortized into an actor-only policy**. Uniform, UCT, AgentUCT, and WTB provide inherited search and execution components (Li et al., 2026). We evaluate the resulting learning protocol, acquisition cost, and transfer.

2 Related Work

Auxiliary evaluation and credit. PPO updates sampled decisions using advantage estimates (Schulman et al., 2017). VinePPO supplies step credit from Monte Carlo state values (Kazemnejad et al., 2025). It is our closest estimator comparison: both credit the main actor trajectory, allowing us to compare state-value credit with ASCT's action table. COMA uses a policy-weighted counterfactual baseline with a centralized critic (Foerster et al., 2018); ASCT estimates values through workflow execution.

Tree-based RL for agentic decisions. BPO uses sibling-relative credit at high-entropy branch states; Tree-GRPO uses intra- and inter-tree advantages; TreePS-RAG derives process supervision from descendants (He et al., 2026; Ji et al., 2026; Zhang et al., 2026). ATPO combines dialogue branching with critic-based credit, and AT^{2}PO uses turn-level tree credit (Cao et al., 2026; Zong et al., 2026). These learners optimize branched rollout collections. ASCT's tree estimates legal-action values at separately sampled actor states; only actor-trajectory decisions enter PPO. Its multi-step branches execute evidence-dependent component choices, linking workflow consequences to this distinct learning interface.

Search allocation and execution reuse. UCT allocates trials through reward and uncertainty (Auer et al., 2002; Kocsis & Szepesvári, 2006). LATS and related methods search for deployed actions (Zhou et al., 2024; Koh et al., 2024). We use AgentUCT's reuse-aware allocation and WTB execution infrastructure (Li et al., 2026) to acquire training credit, then assess both acquisition cost and the learned actor.

3 Problem Formulation

A state s_t contains a task, its executed workflow prefix, and observations. The actor $\pi_\theta(a \mid s_t)$ chooses from a finite legal set $\mathcal{A}(s_t)$. A terminal workflow w receives

$$U(w) = P(w) - \lambda C_{\text{exec}}(w)/C_0, \tag{1}$$

where P is answer quality and C_{exec} is execution cost. The learning objective is expected terminal utility. Recoverable prefixes permit alternative continuations at actor-visited states. We hold the PPO learner fixed to compare credit estimators, with B terminal trials per evaluated state. Equal logical budgets can require different physical work.

Auxiliary work is logged separately from workflow execution utility; RQ2 defines the cost-adjusted metrics.

4 ASCT

An agentic decision example. At a retriever choice, the actor may favor a familiar retriever. Another legal retriever could expose evidence that makes an extra retrieval round useful; its value emerges after continuation, stopping, and reranking. ASCT restores the shared prefix and evaluates these multi-step outcomes. The actor still executes its own sampled action; search supplies the comparison used to train that decision (Figure 1).

Separate behavior from evaluation. At visited state s, the frozen actor computes $p_a = \pi_{\text{old}}(a \mid s)$ and samples a_t. An independent random stream evaluates alternatives from recoverable copies of s. Search allocation can change while the actor advances with a_t.

Retain multi-step action values. Tree nodes are workflow prefixes and edges are legal actions. Trials execute selected paths to completion and back up U. Root coverage evaluates every action

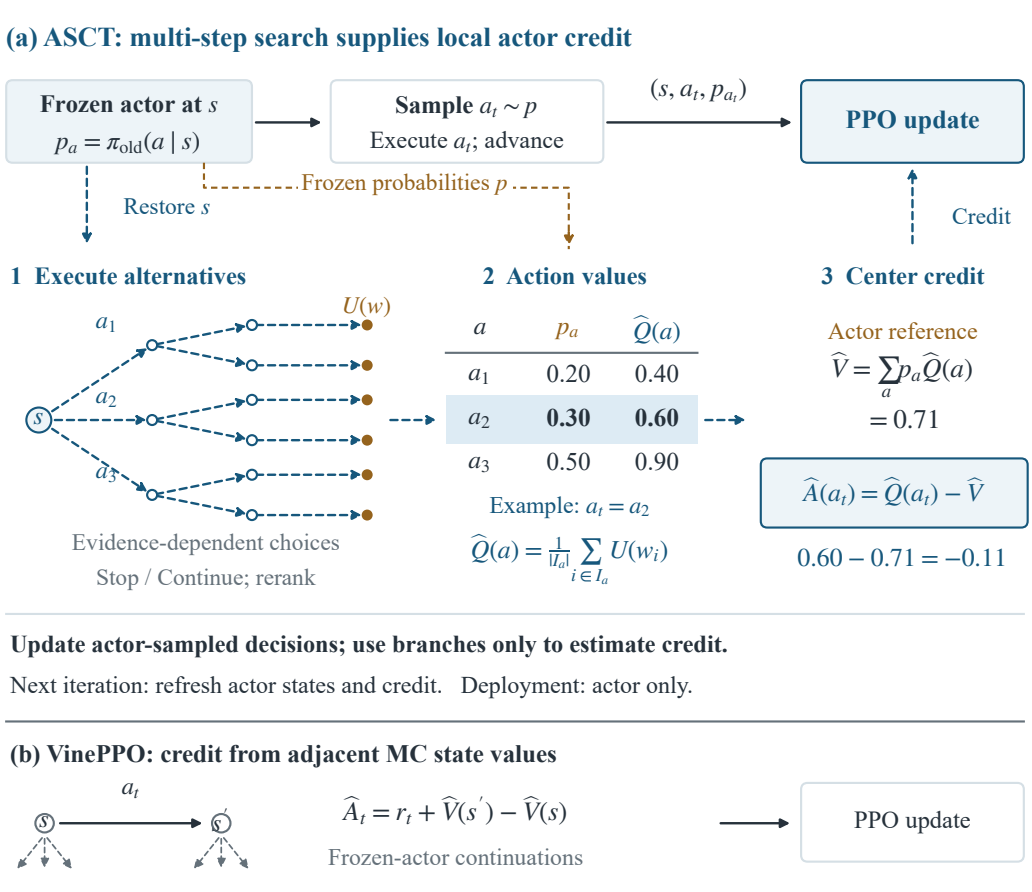

Figure 1: ASCT action-value credit and VinePPO state-value credit share actor-trajectory PPO updates. Solid arrows carry behavior records; dashed arrows carry auxiliary information. Frozen probabilities center ASCT's action table. Values and tree geometry are illustrative; subscripts are suppressed.

before repeat allocation ($B \geq |\mathcal{A}(s)|$). Each value averages the returns of its root-action trials:

$$\widehat{Q}_{E,B}(s,a) = \frac{1}{|\mathcal{I}_{E,B}(s,a)|} \sum_{i \in \mathcal{I}_{E,B}(s,a)} U(w_i). \tag{2}$$

Here $\mathcal{I}_{E,B}(s,a)$ indexes trials starting with a; evaluator E and budget B determine their suffixes.

Credit the sampled action relative to the actor.

$$\widehat{V}_E(s) = \sum_{a \in \mathcal{A}(s)} p_a \widehat{Q}_{E,B}(s,a), \tag{3}$$

$$\widehat{A}_E(s,a_t) = \widehat{Q}_{E,B}(s,a_t) - \widehat{V}_E(s). \tag{4}$$

For fixed $\widehat{Q}_{E,B}$, let $L_s(\theta) = \sum_a \pi_\theta(a \mid s)\widehat{Q}_{E,B}(s,a)$. At $\pi_\theta = \pi_{\text{old}}$, its derivative with respect to softmax logit z_a is $\partial L_s / \partial z_a = p_a \widehat{A}_E(s,a)$. Thus, actor-centered credit provides a local policy-gradient signal before normalization and clipping.

Reuse execution and update online. Uniform balances visits; UCT uses utility and exploration; AgentUCT also accounts for predicted uncached cost. WTB restores shared prefixes and executes uncached suffixes for all evaluators and VinePPO. Each actor decision contributes $(s, a_t, p_{a_t}, \widehat{A}_E(s,a_t))$. After an iteration, credit is standardized and PPO uses $\rho_\theta = \pi_\theta(a_t \mid s)/p_{a_t}$ (Schulman et al., 2017). One frozen actor defines sampling, baseline weights, and the ratio denominator; auxiliary branches

supply credit without entering the update trajectories. The next iteration refreshes visited states and credit with the updated actor. Deployment runs the actor alone. Algorithm 1 and Appendix A specify the full procedure and objective; Appendix D illustrates evaluator rules.

5 EXPERIMENTS

Four RQs evaluate final-policy effectiveness, auxiliary cost, cross-dataset transfer, and description-guided choice. Methods share the workflow and learning protocol. Section 6 relates these policy results to supplementary evaluation controls.

5.1 AGENTIC WORKFLOW AND RAGSPACE

We instantiate the workflow in RAGSpace on HotpotQA questions and supplied candidate passages (Yang et al., 2018; Li et al., 2026). The actor successively chooses query formulation, retrieval, evidence selection, continuation, and reranking (Table 1). Retrieval and generation consume passage text; answer annotations score completed workflows. The shared generator and retrieval components remain frozen.

The query choices are the original question, Query2Doc-style expansion (Wang et al., 2023), and two-query decomposition. Retrievers are BM25 (Robertson & Zaragoza, 2009), E5-base-v2 (Wang et al., 2022), BGE-M3 hybrid (Chen et al., 2024), Qwen3-Embedding-0.6B (Zhang et al., 2025), and reciprocal-rank fusion of their rankings (Cormack et al., 2009). Passage selection uses relevance or maximal marginal relevance (Carbonell & Goldstein, 1998); optional reranking uses Qwen3-Reranker-0.6B (Zhang et al., 2025).

Table 1: Executable RAG decisions. A workflow uses up to three retrieval rounds. Choices within a row are legal alternatives at that stage.

Stage	**Actions**
Query formulation	Original question; Query2Doc; two-query decomposition
Retriever	BM25; E5 dense; BGE-M3 hybrid; Qwen3 embedding; reciprocal-rank fusion
Passage selection	Relevance ranking; maximal marginal relevance
Retrieval width	3 or 6 passages
Retrieval control	Stop; continue while fewer than three rounds have executed
Reranking	Preserve order; Qwen3 reranker
Answer context	Top 2 or top 4 evidence passages

Evidence-dependent control. After retrieval, Stop ends acquisition; Continue refines the query from retrieved evidence, repeats the chosen retrieval rule, and merges passages. The actor then chooses again. At round three only Stop is legal; reranking and context-width choices finish the workflow. Thus earlier choices change later evidence and the value of continuing. Prompts and deterministic refinement are fixed across methods (Appendix A.3).

Actor and observations. The planner sees the question, stage, selected components, round, active queries, evidence titles, and legal labels. It scores legal action responses and normalizes their probabilities; training samples actions and held-out evaluation is greedy. The actor is Qwen3-4B-Instruct-2507 (Qwen Team, 2025) with LoRA (Hu et al., 2022); query and answer generation use the frozen backbone with its adapter disabled. Learning changes workflow decisions while keeping component implementations fixed.

Protocol and comparisons. All methods share the actor, workflow, splits, and PPO configuration (batch size eight; two update epochs per iteration). We train for three iterations on 2,000 questions and evaluate final checkpoints on 400 test questions over seeds 11, 23, and 37. Training uses the repository F1 scorer; endpoint reporting uses official F1, with $\lambda = 0.1$ and $C_0 = 4096$. VinePPO estimates actor-continuation state values for one-step, unit-discount credit; PPO uses terminal trajectory credit. ASCT-Uniform, UCT, and AgentUCT vary the evaluator. VinePPO and ASCT share $B = 12$ and WTB reuse. Logical budgets and realized work are reported separately. We set $c_{\text{exp}} = 1.4$ and $c_{\text{tok}} = \beta = 10^{-4}$; Appendix A gives the full configuration.

5.2 RQ1: Does counterfactual credit improve the final policy?

Table 2: HotpotQA held-out results of the final policies. Workflow utility is the mean over test questions, using official F1 and execution words. Trained rows are mean ± sample SD over three seeds.

Method	Workflow utility	Answer F1	Execution words
Base	0.5648	0.6954	5,348.7
PPO	0.5651 ± 0.0021	0.6913 ± 0.0021	5,170.4 ± 76.8
VinePPO	0.5939 ± 0.0062	0.7115 ± 0.0026	4,817.7 ± 179.7
ASCT-Uniform	**0.6146 ± 0.0191**	**0.7316 ± 0.0168**	**4,792.2 ± 151.4**
ASCT-UCT	**0.6075 ± 0.0241**	**0.7257 ± 0.0260**	**4,841.2 ± 112.8**
ASCT-AgentUCT	**0.6187 ± 0.0076**	**0.7266 ± 0.0126**	**4,420.8 ± 213.3**

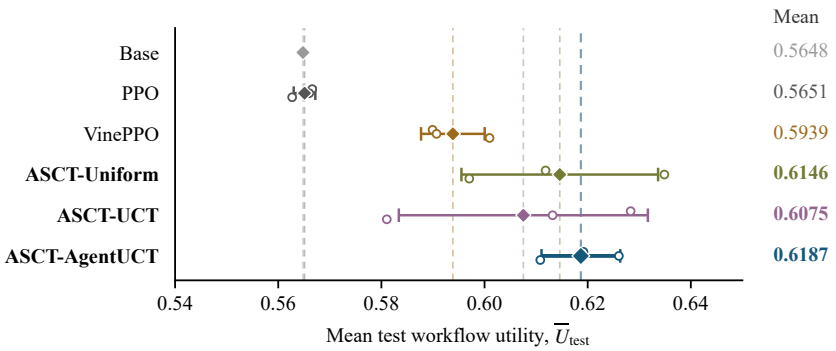

Figure 2: HotpotQA test utility. Diamonds and dashed guides mark means; bars show ±1 sample SD over three seeds; open circles show seeds. Base is fixed; ASCT labels are bold.

All three ASCT evaluators improve mean held-out utility over VinePPO and trajectory-return PPO (Table 2, Figure 2). ASCT-AgentUCT reaches 0.6187, a gain of 0.0248 over VinePPO: 0.0151 comes from higher answer F1 and 0.0097 from lower execution cost. Execution words decrease by 8.24%. The paired utility differences are positive at all three seeds: 0.0293, 0.0354, and 0.0098.

ASCT-Uniform reaches utility 0.6146 and the highest mean F1, 0.7316, supporting the interface with balanced evaluation. The smaller AgentUCT–Uniform difference, 0.0041 ± 0.0116, changes sign across seeds; acquisition cost also matters to evaluator choice.

Where the gains occur. All ASCT policies improve medium- and hard-question utility over VinePPO and PPO (Figure 3). AgentUCT gains 0.0307 and 0.0339 over VinePPO, respectively; the easy group is closely matched, with VinePPO slightly higher.

5.3 RQ2: What does auxiliary evaluation cost, and when is it valuable?

Auxiliary tokens are $T_{\text{aux}} = T_{\text{env}} + T_{\text{actor,aux}}$: recorded Qwen environment calls and auxiliary actor scoring after reuse; E5, BGE-M3, and index construction are outside this token sum. For K logical terminal trials and an assumed N subsequent policy uses, define

$$\overline{J}_{\text{search}} = \left[\sum_{i=1}^{K} U(w_i) - \beta T_{\text{aux}}\right]/K, \tag{5}$$

$$J_{\text{deploy}}(N) = \overline{U}_{\text{test}} - \beta T_{\text{aux}}/N. \tag{6}$$

The first measures acquired outcomes after their cost; the second projects auxiliary-cost amortization, assuming held-out utility represents future use. Search backup and policy credit use U. Main-trajectory collection and optimizer scoring have separate counters. The common token weight β defines an accounting convention; GPU time or API price requires model-specific calibration (Appendix B).

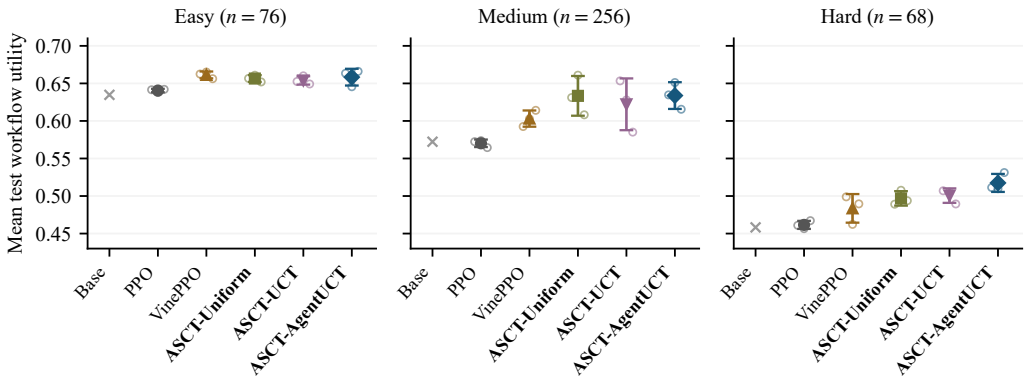

Figure 3: Test utility by difficulty: mean ± sample SD over three seeds, with open seed markers. Base is fixed. Panels share the utility scale and method order.

Auxiliary tokens for the recorded Qwen operations. Table 3 separates environment execution from actor scoring. VinePPO uses 137.09 million environment tokens and 251.29 million tokens to score auxiliary actor continuations, totaling 388.37 million. ASCT obtains continuation decisions from its search rules and incurs no auxiliary actor-scoring calls. AgentUCT uses 193.09 million tokens in this auxiliary ledger, 50.3% below VinePPO and 2.02–2.28% below the other ASCT evaluators. The within-ASCT reductions occur in all three seeds. The 50.3% difference chiefly reflects avoided actor-continuation scoring; VinePPO uses fewer environment tokens. The cost-aware evaluator contributes the smaller within-ASCT reduction.

Table 3: Combined token ledger for the listed Qwen model operations across three training iterations, in millions (mean ± sample SD over three seeds). The first row sums auxiliary environment execution and auxiliary actor scoring within each seed. Indented rows are components of their section subtotal.

Token count (M)	**VinePPO**	**ASCT-Uniform**	**ASCT-UCT**	**ASCT-AgentUCT**
Auxiliary total	**388.374 ± 4.694**	**197.077 ± 0.024**	**197.588 ± 0.068**	**193.088 ± 0.404**
Auxiliary environment execution				
Environment subtotal T_{env}	137.088 ± 1.231	197.077 ± 0.024	197.588 ± 0.068	193.088 ± 0.404
Generator input (4B)	89.972 ± 0.278	139.452 ± 0.162	139.613 ± 0.193	137.776 ± 0.174
Generator output (4B)	3.590 ± 0.009	5.874 ± 0.003	5.883 ± 0.002	5.833 ± 0.003
Query embedding (0.6B)	0.611 ± 0.012	0.689 ± 0.002	0.690 ± 0.001	0.689 ± 0.002
Reranker input (0.6B)	42.916 ± 1.452	51.062 ± 0.152	51.401 ± 0.148	48.789 ± 0.227
Actor scoring (4B)				
Actor-scoring subtotal	305.234 ± 3.515	76.325 ± 0.038	76.208 ± 0.051	76.115 ± 0.148
Auxiliary continuations	251.286 ± 3.472	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
Main-trajectory collection	3.151 ± 0.013	25.442 ± 0.013	25.403 ± 0.017	25.372 ± 0.049
Optimizer forwards	50.796 ± 0.044	50.883 ± 0.025	50.805 ± 0.034	50.744 ± 0.098

The main J metrics use the auxiliary total: environment subtotal plus auxiliary-continuation scoring. Main-trajectory collection and optimizer forwards contribute to the actor-scoring subtotal.

VinePPO reuses frozen action probabilities within each question; scoring is charged once to the first requesting path.

Cost-adjusted evaluation value. Table 4 aggregates rollout utility, auxiliary tokens, and $\overline{J}_{\text{search}}$ over complete runs. Figure 4 shows per-iteration outcomes within ASCT. AgentUCT has the highest mean J_{search} in each iteration and the lowest full-run auxiliary cost. Both use environment execution plus auxiliary actor scoring.

Table 4: Mean auxiliary rollout utility, recorded auxiliary token cost, and $\overline{J}_{\text{search}}$ over complete training runs. Rollout utility uses the training scorer; tokens include the listed Qwen environment operations and auxiliary actor scoring. Values are mean ± sample SD across three seeds.

Evaluator	Mean rollout U	Auxiliary tokens (M)	$\overline{J}_{\text{search}}$
VinePPO	0.5507 ± 0.0027	388.374 ± 4.694	0.4893 ± 0.0019
ASCT-Uniform	**0.5283 ± 0.0022**	**197.077 ± 0.024**	**0.4972 ± 0.0022**
ASCT-UCT	**0.5515 ± 0.0031**	**197.588 ± 0.068**	**0.5203 ± 0.0031**
ASCT-AgentUCT	**0.5526 ± 0.0016**	**193.088 ± 0.404**	**0.5220 ± 0.0016**

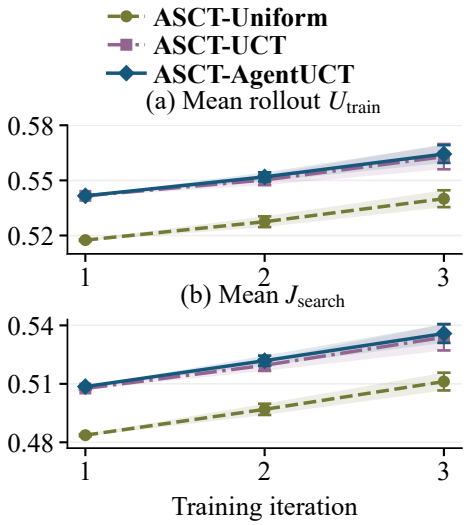

Figure 4: Per-iteration auxiliary evaluation within ASCT: mean rollout U_{train} and J_{search} (2,000 questions; $B = 12$). Utility uses the training scorer. Bands and bars: ± sample SD across three seeds.

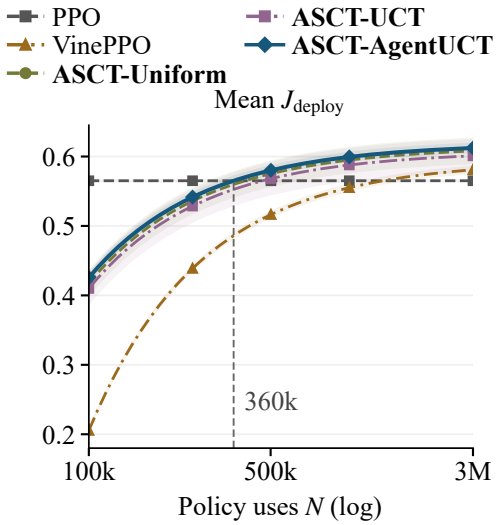

Figure 5: Amortized value across assumed policy uses N (Equation 6; $\beta = 10^{-4}$). Bands: ± sample SD across three seed projections. The guide marks AgentUCT's mean crossing with PPO. Both J metrics charge the recorded auxiliary ledger.

Value after amortizing auxiliary work. Figure 5 projects final-policy value under Equation 6. PPO leads at 100,000 uses; AgentUCT leads at 500,000 and 1,000,000. AgentUCT has higher mean utility and lower auxiliary tokens than VinePPO, giving a mean advantage for every positive N. Its mean curve crosses PPO at approximately 360,358 uses. These thresholds assume the specified token weight and subsequent-use utility; only auxiliary expenditure is amortized.

5.4 RQ3: Does the learned policy transfer across question distributions?

New question distributions. Without further updates, we evaluate on 100 unseen 2WikiMultihopQA bridge-comparison questions and 100 MuSiQue questions (Ho et al., 2020; Trivedi et al., 2022). MuSiQue contains 34 two-hop, 33 three-hop, and 33 four-hop items. Supplied passages are mapped to the same RAG environment; policies share questions, frozen components, and alias-aware answer scoring.

Transfer differs by dataset (Table 5). On 2Wiki, AgentUCT improves both F1 (0.6700 versus 0.6517) and execution words (2,939.3 versus 3,069.3) over VinePPO, raising utility from 0.5767 to 0.5982. On MuSiQue, its lower execution cost offsets slightly lower F1, yielding utility 0.0048 versus -0.0047. These unchanged actors use the same component library under new questions and evidence.

Table 5: Cross-dataset results on the fixed 100-question cohorts. F1, workflow utility, and execution words are mean ± sample SD over three seeds; Base is deterministic.

Dataset	Method	F1	Utility	Execution words
2Wiki	Base	0.5583	0.4732	3,488.0
2Wiki	PPO	0.5883 ± 0.0100	0.5067 ± 0.0102	3,342.6 ± 130.9
2Wiki	VinePPO	0.6517 ± 0.0375	0.5767 ± 0.0360	3,069.3 ± 133.7
2Wiki	**ASCT-Uniform**	**0.6567 ± 0.0321**	**0.5832 ± 0.0329**	**3,009.9 ± 47.6**
2Wiki	**ASCT-UCT**	**0.6583 ± 0.0522**	**0.5837 ± 0.0508**	**3,055.2 ± 59.8**
2Wiki	**ASCT-AgentUCT**	**0.6700 ± 0.0321**	**0.5982 ± 0.0313**	**2,939.3 ± 73.7**
MuSiQue	Base	0.2198	-0.0215	9,880.9
MuSiQue	PPO	0.2181 ± 0.0029	-0.0184 ± 0.0061	9,686.6 ± 144.0
MuSiQue	VinePPO	0.2248 ± 0.0100	-0.0047 ± 0.0128	9,397.0 ± 156.5
MuSiQue	**ASCT-Uniform**	**0.2192 ± 0.0195**	**-0.0072 ± 0.0232**	**9,274.5 ± 181.5**
MuSiQue	**ASCT-UCT**	**0.2225 ± 0.0084**	**-0.0041 ± 0.0085**	**9,284.5 ± 272.0**
MuSiQue	**ASCT-AgentUCT**	**0.2192 ± 0.0135**	**0.0048 ± 0.0082**	**8,782.3 ± 358.6**

5.5 RQ4: Do trained policies retain description-guided component choice?

A newly available component. A fixed retriever A extends Qwen embedding retrieval with a query instruction and metadata rule. In a 100-question constructed probe, annotation-derived tags mark answer-bearing passages and A prioritizes them. Its implementation, name, corpus, actor weights, and all existing component descriptions stay fixed. Four conditions vary only A's description: type plus the correct evidence relation, type and construction details only, name only, and type plus the opposite relation. Appendix F summarizes the protocol.

Table 6: Selection rate for the same new retriever under four descriptions. The correct relation states a preference for tagged, answer-bearing evidence; the opposite relation states a preference for untagged passages. Values are mean ± sample SD over three seeds.

Planner	Type + correct relation	Type only	Name only	Type + opposite relation
Base	0.820	0.010	0.000	0.020
PPO	0.830 ± 0.044	0.033 ± 0.042	0.000 ± 0.000	0.020 ± 0.000
VinePPO	0.947 ± 0.035	0.280 ± 0.108	0.000 ± 0.000	0.040 ± 0.010
ASCT-Uniform	**0.960 ± 0.017**	**0.300 ± 0.111**	**0.000 ± 0.000**	**0.037 ± 0.006**
ASCT-UCT	**0.933 ± 0.021**	**0.197 ± 0.068**	**0.000 ± 0.000**	**0.027 ± 0.012**
ASCT-AgentUCT	**0.950 ± 0.036**	**0.373 ± 0.205**	**0.000 ± 0.000**	**0.050 ± 0.020**

Selection responds strongly to the description while the component itself stays fixed (Table 6). Under the correct evidence relation, ASCT selects A on 93.3–96.0% of questions, compared with 94.7% for VinePPO and 82.0% for Base. Removing the relation lowers ASCT selection to 19.7–37.3%; name-only descriptions yield zero selection, and the opposite relation yields 2.7–5.0%. The contrasts show that the trained actors retain description-guided choice across this library extension. This is a controlled selection probe with constructed tags; the cross-dataset study separately evaluates answer quality and execution efficiency.

6 Discussion

What the learned-policy results establish. The common gain across Uniform, UCT, and AgentUCT supports ASCT's action-evaluation interface across search rules. Uniform's effectiveness is evidence that the benefit does not require cost-aware allocation. The methodological contribution is the connection from executable multi-step alternatives to actor-trajectory updates; the following controls characterize how evaluation supplies that connection.

Tree allocation and evaluation budget. On common five-action states from two questions and three frozen actors, root MC and Uniform share root coverage, uniform suffix sampling, and WTB reuse. At $B = 12$, Uniform lowers repeat Q SD from 0.1203 to 0.0619 and raises sign agreement from 63.3% to 80.0%, using 1.7% more tokens (Table 15). The Q-SD ordering reverses at $B = 48$. Figure 6 traces information acquisition: at 48 trials, UCT, AgentUCT, and Uniform reach evaluated utilities 0.6464, 0.6584, and 0.5322, with sign agreement 66.7%, 90.0%, and 90.0%. These are repeatability diagnostics; RQ1 tests learned-policy effectiveness. Appendix E supplies costs and full controls.

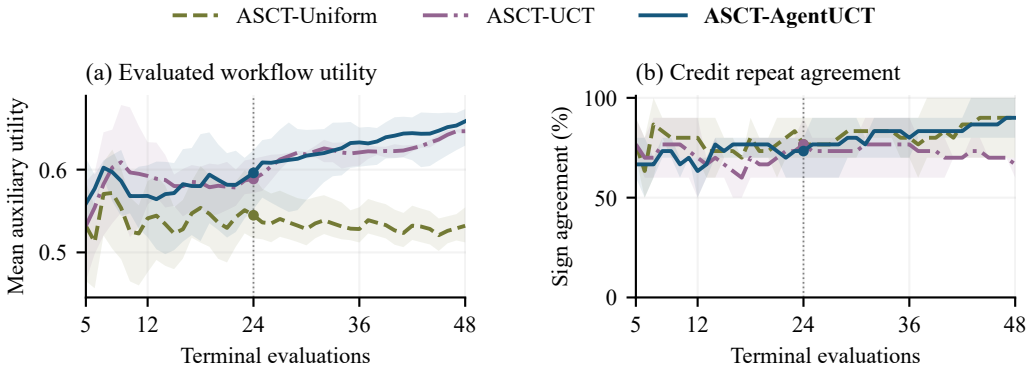

Figure 6: Fixed-state evaluation on two questions, three frozen actor seeds, and two repeats per state. Lines are seed means; bands span seed means. Credit agreement compares signs across repeats. All budgets from 5 to 48 are retained; the guide marks 24.

Actor-rollout evaluation within the interface. With shared actor-centered credit and PPO settings, Uniform improves test utility over ActorRollout at every seed and uses 60.2% fewer auxiliary tokens (Table 7). ActorRollout balances roots and samples frozen-actor suffixes; this contrast tests the full suffix-evaluation mechanism, including allocation and continuation sampling.

Table 7: ASCT continuation control: shared $B = 12$, credit, PPO settings, and 400 test QIDs. Full-run auxiliary tokens; mean $\pm$ sample SD over three seeds.

Continuation variant	Test F1	Test Utility	Aux. tokens (M)
ASCT-ActorRollout	0.7142 ± 0.0090	0.5975 ± 0.0128	495.24 ± 13.29
ASCT-Uniform	0.7316 ± 0.0168	0.6146 ± 0.0191	197.08 ± 0.02

Learning from branch collections. Our AT^{2}PO adaptation performs close to PPO under the shared configuration (Appendix F). Branch records stay fixed within each update and refresh next iteration. This result concerns that adaptation without method-specific tuning.

7 Limitations

Evidence covers one backbone, a finite RAG library, three training seeds, 100-question transfer cohorts, and a constructed description probe. Recoverable execution and finite legal actions delimit applicability. Frozen-state repeatability diagnoses evaluation; it does not isolate each component's contribution to policy gains. Cost projections use the stated auxiliary-token convention; time and price require calibration.

8 Conclusion

ASCT turns multi-step workflow comparisons into actor-centered credit for PPO on actor-sampled decisions. Three evaluators improve mean held-out utility; cost accounting and transfer studies characterize these gains. The learned policy executes without deployment search.

REPRODUCIBILITY STATEMENT

The experiment package retains per-question results, QID manifests, training ledgers, compact credit records, and fixed-state search traces. Appendices A–G document their scope and link reported quantities to the retained artifacts. Paper sources and vector figures accompany the manuscript. Recomputing retained results and reproducing training are separate procedures; training additionally requires the recorded models, datasets, and runtime dependencies.

AI USE STATEMENT

Following the authors' directions, generative AI tools assisted with (1) identifying related work, (2) implementing experimental methods and code, (3) drafting manuscript sections and explanatory figures, and (4) revising wording. They also provided feedback on experimental methodology and interpretation of results, and assisted with reference, formatting, and code-test checks. The authors take responsibility for the final submission and its claims.

REFERENCES

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002. doi: 10.1023/A:1013689704352. URL https://homes.di.unimi.it/~cesabian/Pubblicazioni/ml-02.pdf.

Ruike Cao, Shaojie Bai, Fugen Yao, Liang Dong, Jian Xu, and Li Xiao. ATPO: Adaptive tree policy optimization for multi-turn medical dialogue. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2603.02216.

Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–336, 1998. doi: 10.1145/290941.291025.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-Embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2318–2335, 2024. URL https://aclanthology.org/2024.findings-acl.137/.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 758–759, 2009. doi: 10.1145/1571941.1572114.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *AAAI Conference on Artificial Intelligence*, 2018. URL https://arxiv.org/abs/1705.08926.

Bowei He, Yankai Chen, Xiaokun Zhang, and Xue Liu. Branching policy optimization: Sandbox-native language agent reinforcement learning. *arXiv preprint arXiv:2607.14171*, 2026.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *International Conference on Computational Linguistics*, 2020. URL https://aclanthology.org/2020.coling-main.580/.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. Tree search for LLM agent reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.21240.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. VinePPO: Refining credit assignment in RL training of LLMs. In *International Conference on Machine Learning*, 2025. URL https://proceedings.mlr.press/v267/kazemnejad25a.html.

Levente Kocsis and Csaba Szepesvári. Bandit based Monte-Carlo planning. In *European Conference on Machine Learning*, pp. 282–293, 2006. doi: 10.1007/11871842_29.

Jing Yu Koh, Stephen McAleer, Daniel Fried, and Ruslan Salakhutdinov. Tree search for language model agents. *arXiv preprint arXiv:2407.01476*, 2024. URL https://arxiv.org/abs/2407.01476.

Yang Li, Hai Liu, Dian Shao, Yu Wang, Xiyu Chen, Sergey Volkov, Bozhi Wang, Ziyu Sun, Sihang Liu, Ye Luo, and Xiaowei Zhang. Agent-UCT: Upper confidence bounds applied to trees for agentic workflow optimization with cost-awareness. *arXiv preprint arXiv:2607.24162v2*, 2026.

Qwen Team. Qwen3-4B-Instruct-2507. Official model card, 2025. URL https://huggingface.co/Qwen/Qwen3-4B-Instruct-2507.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 2022. URL https://aclanthology.org/2022.tacl-1.31/.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022. URL https://arxiv.org/abs/2212.03533.

Liang Wang, Nan Yang, and Furu Wei. Query2doc: Query expansion with large language models. In *Conference on Empirical Methods in Natural Language Processing*, pp. 9414–9423, 2023. URL https://aclanthology.org/2023.emnlp-main.585/.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing*, 2018. URL https://aclanthology.org/D18-1259/.

Tianhua Zhang, Kun Li, Junan Li, Yunxiang Li, Hongyin Luo, Xixin Wu, James Glass, and Helen Meng. TreePS-RAG: Tree-based process supervision for reinforcement learning in agentic RAG. *arXiv preprint arXiv:2601.06922*, 2026.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025. URL https://arxiv.org/abs/2506.05176.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning, acting, and planning in language models. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2310.04406.

Zefang Zong, Dingwei Chen, Yang Li, Qi Yi, Bo Zhou, Chengming Li, Bo Qian, Peng Chen, and Jie Jiang. AT^{2}PO: Agentic turn-based policy optimization via tree search. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026. URL https://aclanthology.org/2026.acl-long.1106/.

A CONFIGURATION AND EXECUTION DETAILS

A.1 EVALUATOR AND UPDATE IMPLEMENTATION

Search attention specifies how B trials are assigned to root actions and subsequent branches. **Uniform** selects the least-visited child at fully expanded nodes and samples expansions and continuations uniformly. **UCT** directs repeat visits using mean utility and an exploration bonus, retaining uniform expansion and continuation. **AgentUCT** also accounts for predicted uncached continuation cost. At a fully expanded node v, its selection score is

$$S(v,a) = \overline{U}(v,a) + c_{\exp}\sqrt{\frac{\log(n(v)+1)}{n(v,a)}} - c_{\text{tok}}\widehat{T}(v,a), \tag{7}$$

where $\overline{U}(v,a)$ is the backed-up empirical mean, n denotes visit counts, and $\widehat{T}$ predicts uncached continuation work. The coefficients control exploration and estimated cost. Expansion and continuation sample candidate actions with probability proportional to $\exp[-c_{\text{tok}}\widehat{T}(v,a)]$. The implementation averages predicted remaining cost across legal terminal suffixes. These inherited search rules determine the information acquired for Equations 3–4.

WTB restores shared materialized prefixes and executes only uncached suffixes, allowing counterfactual comparisons to reuse prior workflow computation. AgentUCT additionally uses cache-conditioned cost estimates to allocate trials; all ASCT evaluators and VinePPO share the reuse infrastructure. A partial hit restores the longest cached prefix and executes the remaining suffix. A terminal hit contributes a logical evaluation with zero new environment tokens. Each method, seed, and worker has an isolated cache namespace. The realized ledger measures executed work; the estimated cost guides allocation.

A.2 ONLINE PPO TRAINING AND TREE-FREE DEPLOYMENT

Each iteration collects fresh actor trajectories under π_{old}, constructs their auxiliary credit, and updates the actor. Training standardizes the collected advantages within the iteration. With standardized credit $\widetilde{A}_E$ and ratio $\rho_\theta = \pi_\theta(a_t \mid s)/\pi_{\text{old}}(a_t \mid s)$, PPO maximizes (Schulman et al., 2017)

$$L(\theta) = \mathbb{E}\Big[\min\Big(\rho_\theta \widetilde{A}_E, \text{clip}(\rho_\theta, 1-\epsilon, 1+\epsilon)\widetilde{A}_E\Big)\Big]. \tag{8}$$

The same frozen actor defines action sampling, baseline weights, and the PPO denominator. Thus $\rho_\theta = \pi_\theta(a_t \mid s)/p_{a_t}$ starts at one before updating, avoiding an initial actor–search behavior mismatch. PPO uses actor-sampled trajectories with evaluator-derived credit.

The next iteration evaluates the updated actor's visited states. Online here refers to this alternating collection-and-update procedure. Deployment selects actions from the trained actor and executes the workflow; auxiliary trees are used during training credit construction.

The workflow stages and legal choices appear in Table 1. The following tables specify the common learner and credit estimators.

The main-task data are mutually disjoint subsets of the official HotpotQA training partition: 2,000 training, 400 validation, and 400 test questions. Both held-out sets contain 76 easy, 256 medium, and 68 hard questions to preserve the training-pool difficulty proportions. Final iteration-3 checkpoints are evaluated with greedy actor actions, using the official answer scorer. We report means and sample standard deviations over seeds 11, 23, and 37.

Answer scoring. Training computes multiset token-overlap F1 after case folding and extracting lowercase alphanumeric tokens; articles are retained and there is no special yes/no/noanswer rule. Held-out HotpotQA reporting uses the official scorer: lowercase, punctuation deletion, article removal, whitespace normalization, and zero credit for a mismatched yes/no/noanswer answer. The reward definition used during collection is fixed across methods; official endpoint scores evaluate the resulting answers under the same reporting rule.

Algorithm 1 ASCT with actor-centered credit and PPO

Inputs: actor π_θ, training tasks $\mathcal{D}$, evaluator E, budget B, utility U.

for each training iteration **do**
  Freeze $\pi_{\text{old}} \leftarrow \pi_\theta$; initialize actor-record buffer $\mathcal{R} \leftarrow \varnothing$.
  for each task in $\mathcal{D}$ **do**
    Initialize workflow state s.
    while s is nonterminal **do**
      Compute $p_a = \pi_{\text{old}}(a \mid s)$ for $a \in \mathcal{A}(s)$; sample $a_t \sim p$.
      Initialize an auxiliary tree rooted at a recoverable copy of s.
      Run B tree trials under E, covering all legal root actions first;
      after each trial, back up terminal $U(w)$ and log realized auxiliary tokens.
      Form $\{\widehat{Q}_{E,B}(s,a)\}_{a\in\mathcal{A}(s)}$ from the per-root trial means.
      Set $\widehat{V}_E(s) = \sum_a p_a \widehat{Q}_{E,B}(s,a)$ and $\widehat{A}_E(s,a_t) = \widehat{Q}_{E,B}(s,a_t) - \widehat{V}_E(s)$.
      Append $(s, a_t, p_{a_t}, \widehat{A}_E(s,a_t))$ to $\mathcal{R}$; advance s using a_t.
    end while
  end for
  Standardize advantages in $\mathcal{R}$; update θ with the PPO objective (Eq. 8).
end for; **return** π_θ.

Auxiliary trials use an independent random stream. Policy updates use the sampled actor records in $\mathcal{R}$. Deployment executes the trained actor.

Table 8: Shared training and evaluation configuration. Training uses the repository answer scorer; endpoint reporting uses the corresponding evaluation scorer.

Setting	Value
Planner	Qwen3-4B-Instruct-2507
Adapter	LoRA rank 4, alpha 8, dropout 0; query and value projections
Optimizer	AdamW; learning rate 10^{-5}
Optimizer batch size	8
PPO	Two update epochs per iteration; clipping $\epsilon = 0.2$
Advantage processing	Standardization across collected decisions within each iteration
Train / validation / test	2,000 / 400 / 400 mutually disjoint QIDs
Held-out difficulty counts	76 easy, 256 medium, 68 hard in each set
Seeds and iterations	11, 23, 37; three iterations per trained method
Endpoint rule	Final iteration-3 checkpoint; greedy actor decisions
Auxiliary budget	$B = 12$ terminal trials per actor-visited nonterminal state
Workflow utility	$\lambda = 0.1$, $C_0 = 4096$ execution words
UCT exploration	$c_{\text{exp}} = 1.4$
AgentUCT allocation cost	$c_{\text{tok}} = 10^{-4}$; cache-conditioned suffix estimate
Cost reporting	$\beta = 10^{-4}$ per auxiliary token: environment execution and auxiliary actor scoring

Table 9: Experimental methods. All trained rows use the same planner family and PPO update settings.

Method	Credit at a sampled actor decision	Auxiliary evaluation
Base	No policy update	None
PPO	Terminal trajectory utility	None
VinePPO	Adjacent Monte Carlo state values	Actor continuations
ASCT-Uniform	**Actor-centered action advantage**	**Uniform tree evaluator**
ASCT-UCT	**Actor-centered action advantage**	**Utility-directed UCT evaluator**
ASCT-AgentUCT	**Actor-centered action advantage**	**Cost-aware AgentUCT evaluator**

The planner scores legal action labels and normalizes their probabilities over the current legal set. Collection samples actions from this distribution; evaluation takes its maximum. Advantages are standardized over each iteration's collected decisions before PPO updates. Auxiliary samples update the estimator attached to those decisions.

The workflow execution counter uses lowercase alphanumeric word units. Retrieval charges corpus words for each query; reciprocal-rank fusion charges its four constituent rankings. Maximal marginal relevance and model reranking add the work defined by their passage sets. Generation charges the selected answer-context words. This common execution proxy is distinct from tokenizer-based model-call counts.

Workflow-adapted VinePPO. At each actor-visited nonterminal state, the frozen actor supplies $B = 12$ independently sampled continuations with replacement; their mean terminal utility is the stored state value. Credit is $r_t + \widehat{V}(s_{t+1}) - \widehat{V}(s_t)$ with unit discount. Intermediate rewards are zero; the final actor transition receives its workflow utility and has successor value zero. The stored estimate at the next actor-visited state supplies the preceding transition's successor value, without an additional value-estimation call. Thus B counts auxiliary terminal samples per visited nonterminal state, not per side of a value difference. Main actor execution is separate. Cached terminal outcomes still count as logical samples; cached probabilities are reused, but action draws are fresh. WTB reuse and actor-scoring costs follow the declared ledger.

Legal-action probabilities. For each legal label, the planner scores its JSON action response under the same prompt. The score is the mean conditional log probability over response tokens (excluding prompt and padding tokens). A softmax over these length-normalized scores defines the categorical action distribution. Collection samples this distribution, and PPO uses the corresponding stored normalized action log probability; greedy evaluation selects its maximum. This scoring rule is shared by the compared actors.

A.3 Fixed prompts and query refinement

Table 10 gives the core instructions used across methods. The actor prompt is fixed while its LoRA weights are trained; query and answer generation disable the adapter. The templates are implemented in `experiment_code/src/atppo_lab/policy.py`. The table omits shared JSON-format reminders and retry messages, which remain in the released code.

Table 10: Fixed LLM instructions and their runtime inputs. Quoted instructions are taken from the implementation; output constraints are summarized.

Role / input	Core instruction	Output constraint
Actor / current workflow observation	"You select the next action in a RAG workflow."	JSON with one `action` field; probabilities normalized over legal labels.
Query2Doc / question	"Write a short hypothetical passage that would answer the question."	JSON with one `pseudo_document` string, used as the retrieval query.
Decomposition / question	"Decompose the multi-hop question into exactly two focused search queries."	JSON `queries`: exactly two nonempty strings.
Answer / question and evidence passages	"Answer only from the evidence. Return the shortest answer span that satisfies the question."	JSON `answer` and `citations`; use `UNKNOWN` if evidence is insufficient and cite relevant evidence.

Deterministic refinement. Continue uses `BridgeQueryRefiner` in `experiment_code/src/atppo_lab/rag.py`, with no LLM call. It prioritizes evidence whose title occurs in the question, extracts a bridge entity with fixed text patterns, and falls back to an unseen evidence title, then the first title or the original question. Fixed templates target birth, death, government position, or

location; otherwise the query is "Facts about {entity} needed to answer: {question}." Fixed workflow rules implement the original-question action, Stop/Continue limits, retrieval widths, and context cutoffs.

B TOKEN LEDGER AND COST-ADJUSTED VALUE

The main-text auxiliary-token ledger separates environment execution and auxiliary actor scoring:

$$T_{\text{aux}} = T_{\text{env}} + T_{\text{actor,aux}}. \tag{9}$$

The environment term expands as

$$T_{\text{env}} = T_{\text{4B,in}} + T_{\text{4B,out}} + T_{\text{query embedding,0.6B}} + T_{\text{reranker,0.6B}}. \tag{10}$$

These are realized counts after reuse. Auxiliary actor scoring charges the recorded tokens for evaluating continuation-action probabilities. Main-trajectory actor scoring, optimizer forward passes, E5, BGE-M3, and shared index construction have separate counters. AgentUCT's allocation estimate is a cache-conditioned proxy for remaining generative work; the reporting ledger includes the recorded embedding and reranking calls. All token categories use unit weight under the stated β.

Table 11: Auxiliary evaluation cost within ASCT. Totals are millions of environment tokens. Per-state values normalize within each seed. Search return charges realized auxiliary work as in Equation 5. Values are mean ± sample SD.

Evaluator	Total tokens (M)	Tokens / actor state	$\overline{J}_{\text{search}}$
ASCT-Uniform	197.077 ± 0.024	3,732.5 ± 3.2	0.4972 ± 0.0022
ASCT-UCT	197.588 ± 0.068	3,749.5 ± 2.6	0.5203 ± 0.0031
ASCT-AgentUCT	**193.088 ± 0.404**	**3,664.5 ± 9.2**	**0.5220 ± 0.0016**

Table 3 reports the combined ledger in Section 5.3. Figure 7 shows total auxiliary volume first, followed by the four environment categories for VinePPO and ASCT. PPO has zero auxiliary evaluation tokens; its main-trajectory and optimizer scoring remain part of training. The lower panels use a common method order and an independently labeled, zero-based scale for each category.

For method m and seed r, the projection is computed before aggregation:

$$J_{\text{deploy},m,r}(N) = U^{\text{test}}_{m,r} - \beta T^{\text{aux}}_{m,r}/N. \tag{11}$$

Reported standard deviations are across these seed-level projections. For a comparison with positive mean utility difference $\Delta\overline{U}$ and positive mean token difference $\Delta\overline{T}$, the crossing of mean curves is

$$N_* = \beta\Delta\overline{T}/\Delta\overline{U}. \tag{12}$$

With total auxiliary cost, **ASCT-AgentUCT** versus VinePPO has $\Delta\overline{U} = 0.02481849$ and $\Delta\overline{T} = -195{,}286{,}252.33$ tokens. Higher mean utility and lower mean cost give a positive projected value difference for every $N > 0$.

Table 12: Projected final-policy value after amortizing total auxiliary tokens: $J_{\text{deploy}}(N)$, mean ± sample SD over three seed-level projections. N is the assumed number of uses. These values correspond to Figure 5.

Method	$N = 100{,}000$	$N = 500{,}000$	$N = 1{,}000{,}000$
PPO	0.5651 ± 0.0021	0.5651 ± 0.0021	0.5651 ± 0.0021
VinePPO	0.2055 ± 0.0046	0.5162 ± 0.0056	0.5550 ± 0.0059
ASCT-Uniform	**0.4175 ± 0.0190**	**0.5752 ± 0.0191**	**0.5949 ± 0.0191**
ASCT-UCT	**0.4099 ± 0.0241**	**0.5680 ± 0.0241**	**0.5878 ± 0.0241**
ASCT-AgentUCT	**0.4256 ± 0.0075**	**0.5801 ± 0.0076**	**0.5994 ± 0.0076**

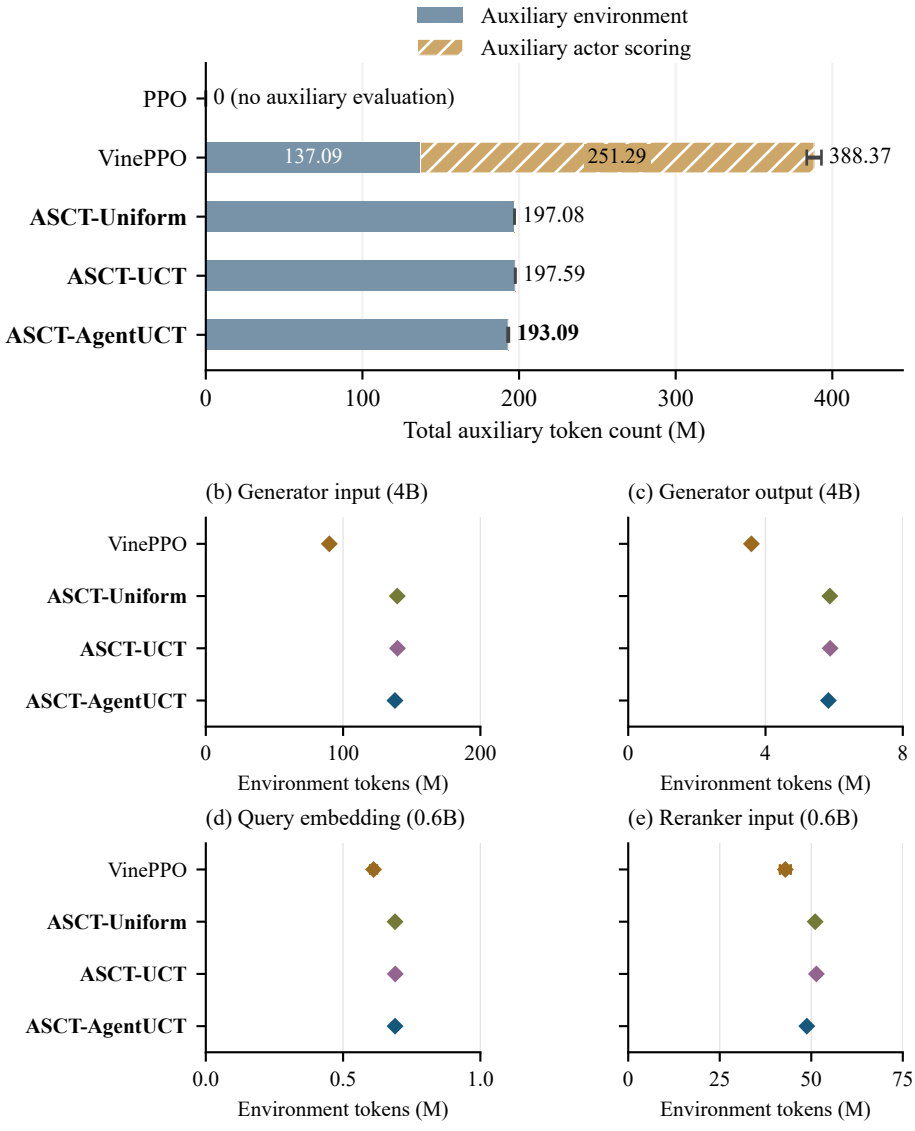

Figure 7: Auxiliary token volume for the recorded Qwen model operations over complete three-iteration training runs. (a) Stacked means sum environment calls and auxiliary actor scoring; endpoint whiskers give the sample SD of each seed's summed count. Numbers at the right are totals. (b–e) Environment categories for VinePPO and the three ASCT evaluators, in the same top-to-bottom order. Each panel has an independent zero-based token axis. Diamonds and whiskers show mean ± sample SD across three seeds. M denotes one million tokens. Boldface identifies all ASCT variants.

C DIFFICULTY-RESOLVED POLICY RESULTS

The test questions are grouped by their supplied HotpotQA difficulty labels. Each method–seed group contains the same 76 easy, 256 medium, and 68 hard questions. Utility uses the official answer scorer and the execution-word penalty in Equation 1. We compute each seed's group mean before reporting the mean and sample SD across seeds.

Table 13: Final-policy utility by HotpotQA test difficulty, using the official answer scorer. Values are mean ± sample SD across three training seeds; Base is deterministic. The three difficulty groups partition the same 400 test questions used in Table 2.

Method	Easy ($n = 76$)	Medium ($n = 256$)	Hard ($n = 68$)
Base	0.6348	0.5723	0.4582
PPO	0.6405 ± 0.0023	0.5702 ± 0.0050	0.4616 ± 0.0053
VinePPO	0.6613 ± 0.0046	0.6031 ± 0.0108	0.4836 ± 0.0191
ASCT-Uniform	**0.6565 ± 0.0045**	**0.6334 ± 0.0265**	**0.4968 ± 0.0097**
ASCT-UCT	**0.6540 ± 0.0056**	**0.6222 ± 0.0345**	**0.5005 ± 0.0097**
ASCT-AgentUCT	**0.6583 ± 0.0111**	**0.6338 ± 0.0178**	**0.5174 ± 0.0120**

D LEARNING INTERFACES, EVALUATOR RULES, AND PAIRED RECORDS

Figure 1 in Section 4 shows how evaluation enters learning. Figure 8 details the allocation rules.

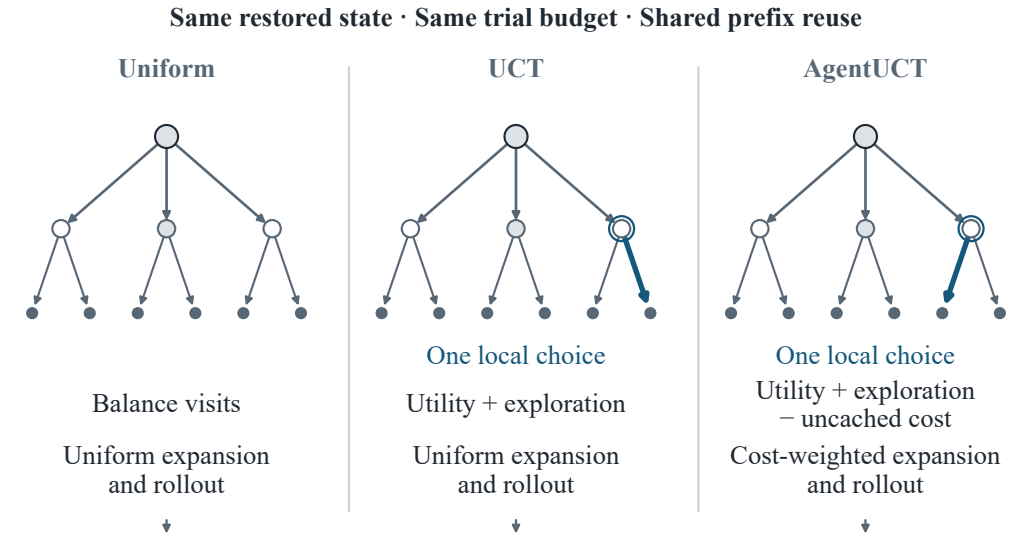

Figure 8: Decision rules within the three ASCT evaluators. UCT and AgentUCT share the depicted tree and prefix; the thick arrow differs only at one circled internal node, illustrating a possible local change in the next choice. Arrow thickness does not encode visit counts or the frequency or magnitude of cost effects. AgentUCT adds predicted uncached cost to selection and uses cost-weighted sampling. All three use WTB prefix reuse, back up terminal utility U, and return root-action means (Li et al., 2026).

Seed-level endpoints and paired utility contrasts are retained in the accompanying data files, including `paired_seed_differences.csv`.

The three training seeds are the units for variability reporting. Methods share the same test questions, enabling paired contrasts. The reported sample standard deviations describe observed variability; the smaller evaluator differences have mixed signs across seeds.

E TRAINING CREDIT AND FROZEN-STATE ALLOCATION CONTROLS

Credit on the observed actor trajectories. We analyze the existing ASCT-Uniform training records for all three seeds: 2,000 training questions across three iterations give 6,000 trajectories per seed and 158,402 decision records in total. Each record contains the sampled action, frozen actor probabilities, legal-action Q table and raw advantage. We reconstruct the actor-weighted baseline and sampled-action advantage from these fields. Keeping the recorded Q table fixed, we also compute the sampled action's advantage under a uniform-action baseline. Table 14 reports the resulting sign differences on states with at least three legal actions. Both compared advantages must have magnitude above 10^{-8} to count a sign change.

Table 14: Baseline semantics on actual ASCT-Uniform training decisions. Counts cover three training iterations per seed. Multi-action states have at least three legal actions. Sign changes compare actor-weighted and uniform-action centering of the same recorded Q table at the actual sampled action.

Seed	All decisions	Multi-action decisions	Sign changes	Rate (%)
11	52,828	12,000	1,043	8.69
23	52,833	12,000	1,021	8.51
37	52,741	12,000	1,053	8.78

Across the three seeds, 3,117 of 36,000 multi-action decisions change sign (8.66%). These are actions that actually entered training, extending the mechanism analysis beyond selected counterfactual examples. The calculation identifies the local comparison supplied by actor-centered credit. Final learning outcomes are evaluated by the complete-method training comparisons in Section 5.2; the alternative baseline here is an offline recomputation.

Root-stratified controls on common five-action states. The fixed-state follow-up uses the original training split's question indices 5 and 6. Each completed iteration-3 ASCT-Uniform actor (seeds 11, 23, and 37) supplies its sampled retriever state, with five legal root actions. Evaluators share its materialized prefix, actor probabilities and sampled action. Every run starts with a fresh WTB namespace and reset transient ranking/reranking caches; WTB reuse remains enabled within the run. Prefix reconstruction and index preparation precede measurement. Each evaluator has two repeats of 48 terminal trials, retaining every intermediate Q table and executed token increment. The 12-trial comparison uses the corresponding prefixes.

Root MC balances root trials and draws each subsequent legal action independently and uniformly. Uniform tree evaluation uses uniform expansion and rollout sampling but balances visits at internal nodes. This comparison tests the allocation structure under the same suffix-sampling rules and cache availability. Internal allocation changes which suffixes are evaluated; it need not estimate the same fixed continuation distribution as independent MC.

Table 15: Internal tree allocation at the training budget ($B = 12$). Both evaluators cover legal root actions, use uniform expansion/rollout sampling, and retain WTB reuse. Two training questions provide five-action retriever states for each of three frozen actors; two repeats per state. Cells report mean $\pm$ sample SD across seed means. Q SD and credit-sign agreement measure repeatability.

Evaluator	Repeat Q SD	Credit agreement (%)	Tokens (k)
Root-stratified MC	0.1203 ± 0.0653	63.3 ± 11.5	11.09 ± 0.13
ASCT-Uniform	0.0619 ± 0.0293	80.0 ± 10.0	11.28 ± 0.51

At 12 trials, repeat Q SD decreases by 48.6% and credit-sign repeat agreement increases by 16.7 percentage points with internal allocation, at 1.7% greater executed token cost. At 48 trials, Uniform's

Table 16: Root-stratified MC and Uniform tree evaluation on the same five-action states. All values are mean ± sample SD across three seed means. Each state has two repeats; budgets are prefixes of the completed 48-trial records.

B	Evaluator	Repeat Q SD	Credit (%)	Tokens (k)
12	Root-stratified MC	0.1203 ± 0.0653	63.3 ± 11.5	11.09 ± 0.13
12	ASCT-Uniform	0.0619 ± 0.0293	80.0 ± 10.0	11.28 ± 0.51
48	Root-stratified MC	0.0368 ± 0.0088	83.3 ± 5.8	38.15 ± 1.03
48	ASCT-Uniform	0.0484 ± 0.0158	90.0 ± 17.3	36.98 ± 1.67

Q SD is 0.0484 versus root MC's 0.0368, while sign agreement is 90.0% versus 83.3%. The observed repeatability benefit applies to the small budget used in training; the ordering depends on budget. Q SD measures variation across repeated evaluations and credit agreement measures sign repeatability; neither uses an accuracy target.

Four-question binary-state corroboration. A separate completed diagnostic uses the first four training questions and the first control and reranker state visited by each of the same three actors: eight binary-action states per seed, 24 checkpoint–state combinations in total. Five evaluators each make four repeats of 12 trials. Root MC balances root trials with uniform suffixes; Actor MC balances roots with frozen-actor suffixes. Two independent 96-trial Actor-MC batches provide a pooled actor-continuation reference. The resulting 528 evaluations retain complete Q tables, terminal paths, utilities and cost counters.

Table 17: Four-question binary-state diagnostic: eight states per actor seed, four repeats at $B = 12$. Order agreement uses two independent 96-trial actor-continuation batches as its reference. Total auxiliary tokens include actor scoring. Values are mean ± sample SD across three seed means.

Evaluator	Repeat Q SD	Ref. order (%)	Tokens (k)
Root-stratified MC	0.0149 ± 0.0221	97.9 ± 3.6	3.90 ± 0.31
ASCT-Uniform	0.0064 ± 0.0093	99.0 ± 1.8	4.16 ± 0.31
ASCT-UCT	0.0093 ± 0.0148	96.9 ± 5.4	4.18 ± 0.32
ASCT-AgentUCT	0.0085 ± 0.0121	99.0 ± 1.8	4.07 ± 0.37
Actor MC	0.0038 ± 0.0040	99.0 ± 1.8	4.78 ± 0.53

Uniform tree evaluation reduces repeat Q SD relative to root MC in every actor seed, from 0.0149 to 0.0064 on average, using 4,158 versus 3,899 environment tokens. The order-agreement ceiling reflects this binary cohort: both reference batches agree on the action order at every state. For two actions, any nondegenerate convex baseline lies between their Q values; comparing actor and uniform baselines therefore cannot reveal a sign difference. The actual multi-action training decisions above supply the relevant centering evidence.

Functional cost-phase controls. On the five-action follow-up states, eight settings independently enable cost awareness in selection, expansion and rollout. The all-off setting is UCT and the all-on setting is native AgentUCT. With Uniform and root MC as controls, the design contains 120 runs and 5,760 terminal trials. For each state, a separate 240-leaf enumeration weights terminal paths by the product of uniform legal-action probabilities, producing a reference for that named continuation distribution. Six enumerations add 1,440 reference terminals. The earlier one-question pilot retains 30 runs and three further enumerations in the evidence package.

At 48 trials, full AgentUCT reduces mean executed tokens by 3.4% relative to UCT and improves credit-sign repeat agreement from 66.7% to 90.0%. At 12 trials its token count is 3.7% higher. The switches therefore affect the executed search and its local comparisons, with effects that depend on phase combinations and budget. All per-trial records are retained, including reference discrepancies and selection changes. The reference describes uniform-continuation values; discrepancies for adaptive evaluators are reference alignment, not universal Q error. These controls make no policy updates.

Table 18: Cost-aware selection, expansion and rollout (S/E/R) on the same two questions and three actors. Bits enable cost awareness at each phase; 000 is UCT and 111 is native AgentUCT. Cells are mean ± sample SD across seed means, with two repeats per state. WTB reuse is enabled throughout.

S/E/R	Tokens@12 (k)	Tokens@48 (k)	Q SD@48	Credit@48 (%)
UCT (000)	11.14 ± 0.63	35.66 ± 1.36	0.0609 ± 0.0150	66.7 ± 5.8
001	10.94 ± 0.66	35.93 ± 0.73	0.0528 ± 0.0128	80.0 ± 20.0
010	11.69 ± 1.55	37.01 ± 1.96	0.0526 ± 0.0123	76.7 ± 15.3
011	11.55 ± 1.29	34.90 ± 2.24	0.0492 ± 0.0219	80.0 ± 10.0
100	11.24 ± 0.57	34.94 ± 2.12	0.0471 ± 0.0197	70.0 ± 17.3
101	11.01 ± 0.50	36.15 ± 0.68	0.0450 ± 0.0074	76.7 ± 5.8
110	11.77 ± 1.52	36.29 ± 1.16	0.0519 ± 0.0029	73.3 ± 23.1
AgentUCT (111)	11.56 ± 1.47	34.46 ± 0.75	0.0503 ± 0.0239	90.0 ± 10.0
Uniform	11.28 ± 0.51	36.98 ± 1.67	0.0484 ± 0.0158	90.0 ± 17.3
Root MC	11.09 ± 0.13	38.15 ± 1.03	0.0368 ± 0.0088	83.3 ± 5.8

Budget curves from completed evaluations. The original training records use 12 terminal evaluations per state and contain endpoint action tables. For longer budget curves, we use the completed 48-trial fixed-state evaluations on two questions already in the same training split (ordered indices 5 and 6). Each of the three iteration-3 actors supplies its observed five-action retriever state. Two repeated searches per method retain every terminal utility, cumulative executed token count, and reconstructed action table. Prefixes of these records provide every budget from 5 to 48, including 24, with no additional environment execution or policy updates. The same materialized prefix and actor probabilities are used across evaluators within each state. Fresh per-evaluation WTB namespaces and reset transient retrieval/reranker caches isolate auxiliary execution; prefix preparation is outside the measured ledger.

At budget K, mean evaluated utility is $K^{-1} \sum_{i=1}^{K} U(w_i)$. Credit repeat agreement is the fraction of legal actions whose centered-credit signs agree between the two independent searches at that budget. It measures repeatability, with final-policy utility assessed separately in the training experiments. These ASCT evaluators use rule-based continuations and add no auxiliary actor-scoring calls, so $T_{\mathrm{aux}}(K) = T_{\mathrm{env}}(K)$. Equation 5 gives $\overline{J}_{\mathrm{search}}(K)$ using $\beta = 10^{-4}$. We average repeats and questions within each seed before reporting seed means. The bands in Figures 6 and 9 span the three seed-level means; they are descriptive ranges.

Table 19: The 24-evaluation checkpoint of the full curves. Values average the same two questions, two repeats and three actor seeds. The complete curves retain seed-level variability.

Evaluator	Mean utility	Tokens	$\overline{J}_{\mathrm{search}}$	Credit agreement (%)
ASCT-Uniform	0.5449	21,077.9	0.4571	76.7
ASCT-UCT	0.5887	20,788.5	0.5021	76.7
ASCT-AgentUCT	**0.5960**	**20,531.3**	**0.5105**	**73.3**

Observed relation. At 24 evaluations, **ASCT-AgentUCT** has the highest mean auxiliary utility and $\overline{J}_{\mathrm{search}}$ among the three evaluators, while its credit repeat agreement is lower (Table 19). At 48 evaluations, UCT and AgentUCT reach mean utilities 0.6464 and 0.6584, compared with 0.5322 for Uniform. Their corresponding $\overline{J}_{\mathrm{search}}$ values are 0.5721, 0.5866, and 0.4551. Credit repeat agreement is 66.7%, 90.0%, and 90.0%, respectively. Thus, directing evaluation toward higher-utility continuations and producing repeatable action comparisons are distinct properties. These exploratory curves describe information acquisition on two existing training questions. The main experiments measure the learned policy's quality and the aggregate execution cost across training.

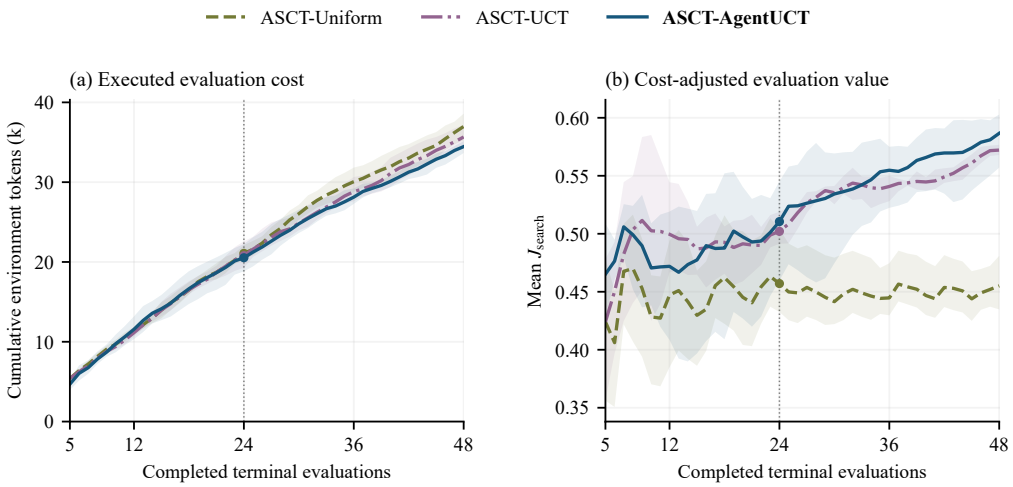

Figure 9: Executed auxiliary tokens and cost-adjusted evaluation value on the same states and budgets as Figure 6. $\overline{J}_{\text{search}}(K) = \overline{U}(K) - 10^{-4}T_{\text{aux}}(K)/K$. Auxiliary actor scoring is zero for these three evaluators, so total auxiliary cost equals environment cost. The deployment amortization curve uses Equation 6. Lines and bands follow Figure 6.

F COMPONENT PROTOCOL AND CONTINUATION DIAGNOSTICS

Description controls. Component A prioritizes passages tagged from answer-bearing annotations in a constructed corpus. The four descriptions vary whether its type and evidence preference are stated correctly, only its construction is stated, no description is given, or the opposite preference is stated. Its implementation, name, corpus, actor weights, and existing component descriptions stay fixed. The correct description also explains the tags, so the comparison does not isolate one wording feature. Per-question answer outcomes are retained in the accompanying data package.

What makes the credit useful? For a fixed table, the local surrogate $L_s(\theta) = \sum_a \pi_\theta(a \mid s)\widehat{Q}_{E,B}(s,a)$ has logit derivative $p_a \widehat{A}_E(s,a)$ at the behavior policy. This explains the direction of the raw credit before normalization and clipping. It benefits actor utility when evaluator preferences align with outcomes the actor can realize. The continuation rule determines which downstream outcomes contribute to each action value. Online recollection refreshes visited states and actor-weighted baselines after each training iteration.

Continuation control. ASCT-Uniform uses uniform tree expansion and continuation with least-visited selection; ASCT-ActorRollout balances root actions and samples suffix actions from the frozen actor. Both retain root coverage and actor-centered credit. This intervention changes the full continuation mechanism, and each condition trains its own evolving actor. Table 7 reports the three-seed held-out endpoints. The matched-QID ASCT-ActorRollout minus ASCT-Uniform test-utility differences are -0.0077, -0.0294, and -0.0143 in seeds 11, 23, and 37. All 18,000 Actor-CF training trajectories were retained; reconstructing the Q-table centering for 158,357 actor decisions found no protocol violations. The audit verifies that recorded credits match the action values and actor-weighted baselines.

Across complete training runs, ASCT-ActorRollout uses 495.24 ± 13.29 million auxiliary tokens: 180.12 million in environment execution and 315.12 million in auxiliary actor scoring on average. Main-trajectory and optimizer actor scoring are excluded, as in the shared ledger. Its mean auxiliary rollout utility is 0.5488.

AT^{2}PO workflow adaptation. We adapted the public tree learner (Zong et al., 2026) to categorical workflow actions under the planned training setup, with batch size eight and three collect–update iterations. Each iteration collects trees with the current actor and optimizes over that collection; the next iteration collects again with the updated actor. Each training question produces a 106-leaf tree whose branch records enter optimization. The training trajectories remain fixed during each update and are refreshed at the next collection.

Independent iteration-3 evaluation on the paper's test QIDs with the official scorer gives test utility 0.5647 ± 0.0042 and F1 0.6945 ± 0.0041 across three seeds. Relative to PPO (0.5651 ± 0.0021), the paired test-utility difference is -0.0004 ± 0.0063. Auxiliary tokens total 696.15 ± 19.53 million per run; optimizer scoring is separate. This is an exploratory evaluation of one adaptation without a method-specific tuning study. The adaptation differs from ASCT in its evaluation unit, credit construction, and policy-update records.

G RETAINED EVIDENCE

The accompanying data package retains matched per-QID endpoints, seed-level auxiliary ledgers, compact credit records, and the component and transfer probes. For the new controls, the compact supplement includes three-seed matched-QID predictions, training metrics, and protocol audits. Full Actor-CF and AT^{2}PO trajectory archives are retained separately by the authors. These artifacts support recomputation of reported results; reproducing training additionally requires the recorded models, indices, and runtime.