TYPED-REWARD-AWARE CREDIT ESTIMATION AND ROUTING FOR MULTI-REWARD AGENTIC REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Practical language-model agents face diverse, task-dependent reward requirements. Common examples concern task outcomes, intermediate processes, and conditions on valid interaction. Reinforcement learning for these agents therefore faces a multi-reward credit-assignment problem, extending beyond distributing a final outcome score across turns. We identify three coupled difficulties: shared temporal propagation can let dense process feedback overwhelm outcome credit; mixed action-type statistics can encourage easily rewarded behaviors; and additive reward fusion can allow task gains to offset violated prerequisites. We propose **TRACER**, Typed-Reward-Aware Credit Estimation and Routing, to preserve these distinctions throughout credit assignment. TRACER estimates reward-specific advantages with separate horizons, baselines, and normalization scales, refines comparisons by action type where appropriate, and uses conditional rewards to gate which pathways enter each update. The normalized pathways are then fused into a single policy advantage. We instantiate three common reward types—outcome, process, and conditional rewards—in search-augmented question answering. Experiments across seven benchmarks and three backbones show improvements in answer accuracy over group-relative baselines. Component analyses connect these gains to auxiliary-reward horizons, action-type comparisons, and sustained format compliance. Together, these results support treating temporal responsibility, action comparability, and conditional eligibility as joint requirements for credit assignment in practical multi-turn agent RL.

1 INTRODUCTION

Reinforcement learning (RL) increasingly trains language-model agents to solve tasks through multi-turn interaction, including search and open-ended research (Jin et al., 2025; Zheng et al., 2025). In practical use, these agents must decide what to retrieve and when to answer within an interaction budget, requiring credit assignment across decisions with delayed consequences.

Recent progress has improved credit assignment from outcome supervision, moving from trajectory-level judgments toward more informative step-level comparisons (Shao et al., 2024; Feng et al., 2025; Fang et al., 2026; He et al., 2026). A broader practical question remains: how should credit be assigned when several distinct requirements must be learned together?

Practical agent training involves diverse, task-dependent reward requirements, spanning task quality, intermediate behavior, tool execution, and adherence to interaction protocols (Wu et al., 2023; Qian et al., 2025; Zhang et al., 2026). Three common reward types are outcome rewards that evaluate task completion, process rewards that judge intermediate behavior, and conditional rewards that encode prerequisites other objectives should not override. These are representative examples within a broader range of reward needs; a task may specify multiple signals of each type or additional reward types. We study *multi-reward credit assignment for practical multi-turn agents*: converting this heterogeneous collection into policy credit while respecting the role of each signal. Jointly handling their different temporal scopes, action statistics, and conditional roles remains an open design challenge. Three difficulties motivate our approach.

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.

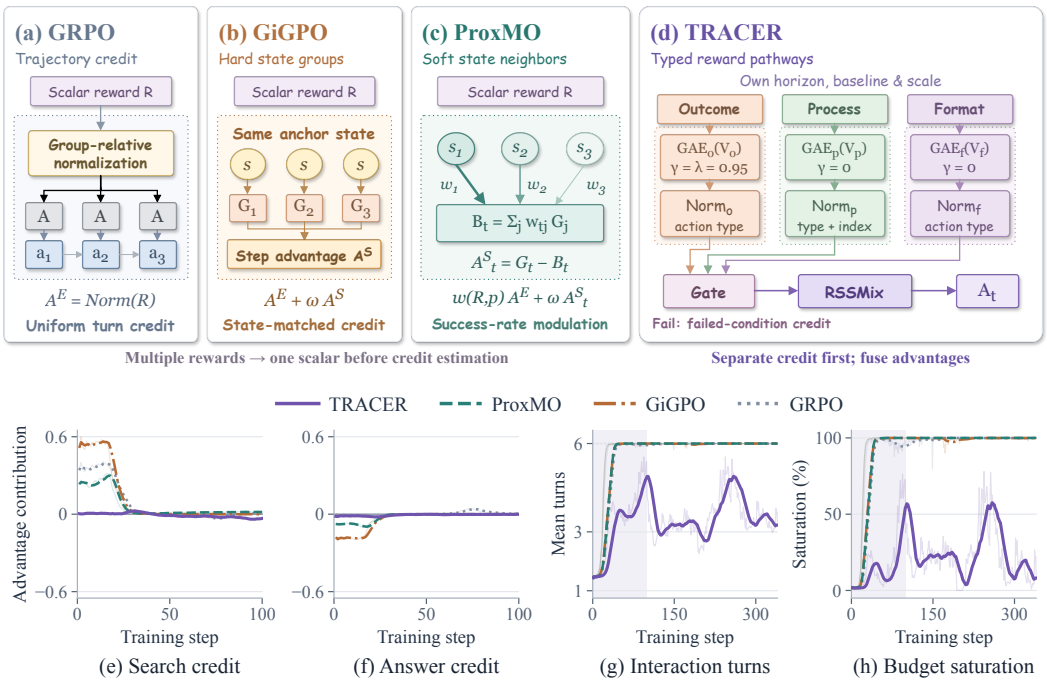


Figure 1: Credit assignment and trajectory behavior on Qwen2.5-1.5B. (a–d) Method comparison. (e–f) Search/answer contributions $C_k = n_k \bar{A}_k/(n_s + n_a)$ (steps 1–100): n_k is the action count, $\bar{A}_k$ the mean final advantage; empty groups contribute zero. (g–h) Mean interaction turns and search-budget saturation (five calls), including subsequent answers (steps 1–340, same runs). Dark/faint lines: trailing means/raw batches (windows: 5 steps in e–f, 20 in g–h); shading: steps 1–100.

Mismatched temporal credit. Process feedback, supplied by a process reward model (PRM) or a rule-based evaluator, can judge a particular reasoning or tool-use step, whereas an outcome reward concerns the completed task. Propagating both over the same horizon can repeatedly accumulate dense process scores and overwhelm outcome credit; cumulative process rewards can also encourage reward hacking (Cheng et al., 2025). In multi-turn interaction, this can favor longer trajectories without better answers. The credit estimator must respect each signal's intended horizon and scale.

Action-type hacking. Different actions can have different reward distributions and difficulty levels. Rewards that overlook these differences can favor easily rewarded action types (Zhang et al., 2026). Mixing such actions in shared comparison pools can further distort their relative credit. In search agents, the corresponding failure can be excessive searching or premature answering. Such fixes require reward calibration across action types, scales, and trajectory lengths.

Conditional requirements reduced to trade-offs. Tool-use protocols impose structural requirements that other rewards should not override. Under additive reward fusion, however, gains in answer quality or process scores can compensate for violating that requirement. The policy may improve task scores while learning the structure incompletely or losing it later in training.

Figure 1 connects these issues to practical training behavior in our search setting, which combines the three common reward types above. Early positive search and negative answer credit contributions in the uncalibrated baseline runs accompany a later drift toward the interaction limit: mean interaction turns increase and nearly every trajectory exhausts the search budget. TRACER maintains more moderate lengths and lower budget saturation. These diagnostics associate credit imbalance with less adaptive tool use; §5.4 examines the separate risk of losing format compliance.

We propose *Typed-Reward-Aware Credit Estimation and Routing* (TRACER) to address these difficulties within a single credit estimator. It preserves reward-specific horizons, baselines, and normalization scales until advantage fusion; uses action-type-scoped comparisons where appropriate; and uses condition gates to retain only failed-condition credit whenever a prerequisite is violated.

These mechanisms reuse the existing trajectory batch and reward evaluators, without training a value network or adding a separate process reward model. Our contributions are threefold:

A practical credit-assignment setting. We identify the joint credit-assignment challenge posed by diverse reward requirements in practical multi-turn agent RL, highlighting mismatched temporal credit, action-type hacking, and conditional requirements that additive rewards can trade off.

A unified credit-assignment framework. TRACER implements these requirements through typed credit pathways, action-type-scoped statistics, and condition-gated fusion, providing a practical interface from heterogeneous rewards to policy updates.

Evaluation across models and requirements. Using outcome, process, and conditional rewards as a representative experimental setting, TRACER improves macro EM over the strongest compared baseline by 1.64, 3.49, and 4.03 percentage points on Qwen2.5-1.5B, Qwen2.5-7B, and Llama-3.1-8B, respectively, across seven search-augmented QA benchmarks, and leads in 20 of 21 backbone–benchmark combinations. Training diagnostics and component analyses connect answer quality to process credit, action selection, and sustained format compliance.

2 Related Work

Multi-turn agents and step-level credit. Reinforcement learning enables language models to learn when to retrieve information, use tools, and terminate an interaction, turning agent behavior into a sequential optimization problem (Jin et al., 2025; Zheng et al., 2025). A central challenge is attributing a trajectory's outcome to its intermediate decisions. **GRPO** (Shao et al., 2024) estimates trajectory-level credit through comparisons among sampled trajectories. Step-level extensions refine which decisions are compared: **GiGPO** (Feng et al., 2025) groups repeated states, **ProxMO** (Fang et al., 2026) uses semantic proximity, and HGPO (He et al., 2026) accounts for historical-context consistency. This progression improves the resolution of credit derived from task outcomes.

Reward requirements in practical agent training. Task success captures only part of what practical agents are expected to learn. Feedback can also assess intermediate reasoning, tool selection, execution details, and adherence to interaction requirements. Fine-grained feedback separates quality dimensions, while process supervision evaluates intermediate reasoning beyond final-answer correctness (Wu et al., 2023; Lightman et al., 2024). In tool-use learning, reward granularity, scale, and timing affect training behavior (Qian et al., 2025); in shopping-behavior modeling, difficulty-aware action rewards help counter a preference for easily rewarded actions (Zhang et al., 2026). These studies motivate diverse reward signals with distinct meanings and optimization roles.

Credit estimation with multiple reward signals. Once several signals are available, their aggregation becomes part of the credit-assignment problem. Reward timing motivates different multi-turn estimators (Wei et al., 2026), while cumulative process rewards can induce hacking even when individual steps receive useful feedback (Cheng et al., 2025). GDPO (Liu et al., 2026) normalizes reward components separately and also studies conditioning one reward on another before normalization; IGPO (Wang et al., 2026) combines separately normalized information-gain and outcome rewards into turn-level returns. TRACER instead makes conditional inclusion decisions on estimated advantages at each action, alongside reward-specific horizons and action-type comparisons. Reward-side conditioning and advantage-side routing are therefore distinct choices, rather than the presence versus absence of conditional rewards.

3 Preliminaries

Agentic trajectories and multiple rewards. For a question q, a policy π_θ samples a group of G trajectories. At turn t of trajectory g, the policy observes history $h_{g,t}$ and generates an action $y_{g,t} \sim \pi_\theta(\cdot \mid h_{g,t})$ of a given *action type*, such as search or answer. We treat each such turn as a transition and assign the t-th transition of the g-th trajectory a set of *immediate rewards* $r_{g,t}^{(k)}$, where $k \in \{1, \ldots, K\}$ indexes a reward signal and its pathway. Throughout, $\epsilon > 0$ is a numerical stabilizer and c_k is a fixed nonnegative weight for the k-th reward. The turn-level scalar advantage $A_{g,t}$ is broadcast to each unmasked response token $y_{g,t,\ell}$ in that action. With token prefix $y_{g,t,<\ell}$,

the PPO clipped surrogate (Schulman et al., 2017) is

$$\mathcal{J}(\theta) = \mathbb{E}_{g,t,\ell}\Big[\min\big(\rho_{g,t,\ell}A_{g,t},\ \mathrm{clip}(\rho_{g,t,\ell}, 1-\epsilon_{\mathrm{clip}}, 1+\epsilon_{\mathrm{clip}})A_{g,t}\big)\Big],$$
$$\rho_{g,t,\ell} = \frac{\pi_\theta(y_{g,t,\ell} \mid h_{g,t}, y_{g,t,<\ell})}{\pi_{\theta_{\mathrm{old}}}(y_{g,t,\ell} \mid h_{g,t}, y_{g,t,<\ell})}. \tag{1}$$

Here ϵ_{clip} is the ratio-clipping range; the actor-loss configuration is specified in Appendix D.1. Our design question is how a set of heterogeneous rewards becomes the single turn-level advantage used to update every valid response token in an agent action.

Group-relative normalization. Group-relative policy optimization (Shao et al., 2024) estimates trajectory credit from relative scores within the G trajectories for the same question. A scalar signal w is centered and scaled within its *comparison pool* $\mathcal{P}_u$, indexed by a label u:

$$z_{g,t} = \frac{w_{g,t} - \mu_u}{\sigma_u + \epsilon}, \qquad \mu_u = \mathrm{mean}_{(g,t)\in\mathcal{P}_u} w_{g,t}, \qquad \sigma_u = \mathrm{std}_{(g,t)\in\mathcal{P}_u} w_{g,t}. \tag{2}$$

The comparison-pool label u is a key degree of freedom in this paper: it decides which transitions a given transition is compared against, and can therefore be set by action type, or by a transition's position within the trajectory, rather than fixed to the whole question group.

Generalized advantage estimation. Given immediate rewards and a state baseline V, generalized advantage estimation (GAE) (Schulman et al., 2016) combines temporal-difference residuals $\delta_t = r_t + \gamma V_{t+1} - V_t$ as $A_t = \sum_{l\geq 0}(\gamma\lambda)^l\delta_{t+l}$, with the sum ending at termination. The discount γ defines the return horizon, while λ weights multi-step TD residuals: $\lambda = 0$ gives one-step TD credit and $\lambda = 1$ gives the discounted Monte Carlo residual for a complete trajectory with a zero terminal baseline. TRACER applies GAE to each reward pathway with group-relative state baselines.

4 METHOD

TRACER keeps each reward signal in a separate *credit pathway* until advantage fusion. Its design follows the three requirements: pathway-specific GAE determines how far feedback propagates (§4.1), action-type-scoped baselines and normalization determine which decisions are comparable (§4.2), and condition-gated routing determines which credit may contribute (§4.3). The resulting pathways are fused with a trajectory anchor into the policy advantage (§4.4). Figure 2 summarizes the computation; Section 3 introduces the trajectory and reward notation.

4.1 PER-PATHWAY GENERALIZED ADVANTAGE ESTIMATION

For each pathway k, let $r^{(k)}$ be its immediate reward. Pathway-specific parameters (γ_k, λ_k) and a group-relative state baseline $V^{(k)}$ (§4.2) give the raw GAE advantage

$$a^{(k)}_{g,t} = \sum_{l=0}^{T_g-t-1} (\gamma_k\lambda_k)^l\,\delta^{(k)}_{g,t+l}, \qquad \delta^{(k)}_{g,t} = r^{(k)}_{g,t} + \gamma_k V^{(k)}_{g,t+1} - V^{(k)}_{g,t}. \tag{3}$$

Here T_g is the trajectory length and $V^{(k)}_{g,T_g} = 0$. The discount γ_k determines the return horizon, while the standard GAE parameter λ_k weights future TD residuals. Outcome credit can therefore propagate across earlier decisions, while local process credit with $\gamma_k = 0$ reduces to $r^{(k)} - V^{(k)}$. This locality concerns temporal propagation within a pathway; normalization and the trajectory anchor still use information from other transitions. Equation 3 assumes a complete sequence with available baselines; missing values or successors interrupt the recursion as specified in Appendix B.2.

4.2 ACTION-TYPE-SCOPED COMPARISON AND NORMALIZATION

Temporal horizons specify which rewards enter a return; comparison pools specify whose returns provide the reference. TRACER estimates baselines within each reward pathway and normalizes its

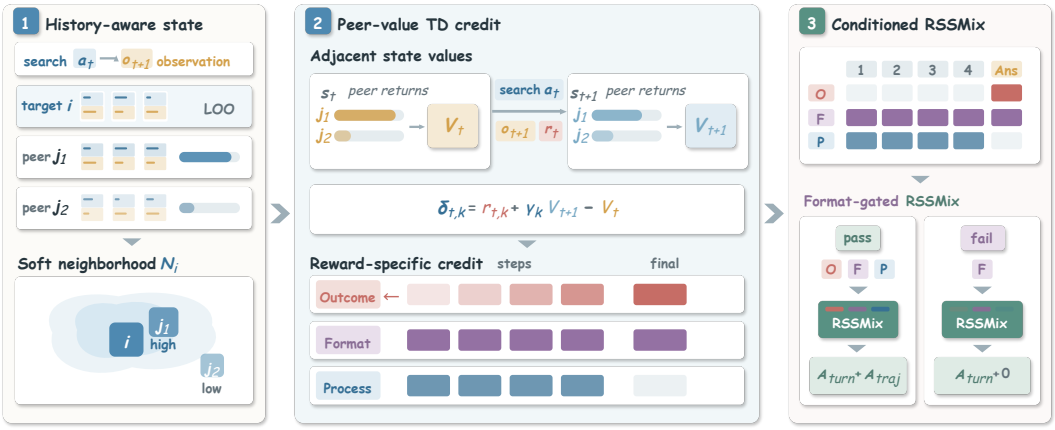


Figure 2: TRACER with leave-one-trajectory-out (LOTO) values, per-pathway GAE, and condition-gated fusion. *Process* denotes both process pathways. Failed gates retain failed-condition credit and suppress the anchor; RSSMix and anchor fusion yield the policy advantage $A_{g,t}$.

advantages before fusion, retaining the reward-wise treatment of scale (Liu et al., 2026). For each pathway, a comparison-pool label refines the question group according to the reward's semantics:

$$u_{g,t}^{(k)} = \big(\,\text{question}(g),\ \phi_k(g,t),\ \tau_k(g,t)\,\big), \tag{4}$$

where ϕ_k identifies the action type and τ_k identifies the turn index; a disabled refinement uses a shared label. Action-type refinement separates search from answer comparisons, preventing their return scales from determining each other's baselines and normalization. Turn-index refinement compares decisions at the same turn index when reward distributions depend on interaction progress.

Let $\mathcal{P}_x^{(k)}$ contain transitions sharing label $u_x^{(k)}$. For $x = (g,t)$, eligible value candidates satisfy $\mathcal{C}_x^{(k)} \subseteq \{(g',t') \in \mathcal{P}_x^{(k)} : g' \neq g\}$, giving leave-one-trajectory-out baselines. Using the pathway return $G_{g,t}^{(k)} = r_{g,t}^{(k)} + \gamma_k G_{g,t+1}^{(k)}$ with $G_{g,T_g}^{(k)} = 0$, the baseline is

$$V_x^{(k)} = \sum_{y \in \mathcal{C}_x^{(k)}} w_{xy}^{(k)} G_y^{(k)}, \qquad w_{xy}^{(k)} = \frac{\exp(K_{xy}/\tau_{\text{sim}})}{\sum_{z \in \mathcal{C}_x^{(k)}} \exp(K_{xz}/\tau_{\text{sim}})}. \tag{5}$$

Here K_{xy} compares interaction histories strictly before the current actions, and $\tau_{\text{sim}} > 0$ is the similarity temperature. Permitted fallbacks relax turn-index matching while preserving question, action type, and exclusion of the entire target trajectory. Appendix B.1 specifies the history kernel, pathway eligibility, and fallback rules. The search implementation uses root-mean-square (RMS) normalization within each pathway pool:

$$z_{g,t}^{(k)} = \frac{a_{g,t}^{(k)}}{\sqrt{\text{mean}_{x \in \mathcal{P}_{g,t}^{(k)}}[(a_x^{(k)})^2] + \epsilon}}. \tag{6}$$

Only valid, applicable credits enter these normalization statistics. Normalization retains the original pool even when value estimation uses a turn-index fallback. RMS scaling preserves the raw credit's sign without imposing zero mean; a pool with zero or nonfinite scale contributes zero. This normalization does not equalize realized gradient contributions across pathways.

4.3 Condition-Gated Advantage Routing

In practical agent training, structural constraints such as format compliance are *preconditions* for credit from other objectives: format precedes content. Each such signal is represented by a *conditional pathway* with its own threshold θ_k. A valid transition passes if the condition is inapplicable

or its immediate reward meets the configured threshold:

$$\begin{aligned} \mathrm{pass}_{g,t}^{(k)} &= \mathrm{valid}_{g,t} \wedge \big(\neg \mathrm{applies}_{g,t}^{(k)} \ \vee \ r_{g,t}^{(k)} \geq \theta_k\big), \\ \mathrm{gate}_{g,t} &= \mathrm{valid}_{g,t} \wedge \bigwedge_{k \in \mathcal{C}} \mathrm{pass}_{g,t}^{(k)}, \end{aligned} \tag{7}$$

where $\mathcal{C}$ is the set of conditional pathways and valid excludes padding and truncated transitions. The gate defines a per-transition, per-pathway *inclusion mask* $m_{g,t}^{(k)}$ over applicable pathways with valid advantages. If the gate passes, all such pathways are included; if it fails, *only* the failed conditional pathways are included, while every other pathway is dropped. On failure, outcome and process pathways contribute zero; only failed-condition credit remains eligible.

This is *multiplicative* routing: a binary mask selects normalized advantages at fusion. The threshold is evaluated on the immediate conditional reward; the mask is applied after GAE and normalization, without modifying their reward inputs or comparison statistics. The trajectory anchor is also zeroed at a failing transition (§4.4), preventing outcome credit from entering through that channel. The retained conditional advantage is relative and need not be negative; routing does not guarantee constraint satisfaction. Appendix A.5 gives the exact inclusion rule and its search instantiation.

4.4 Advantage fusion and trajectory anchoring

RSSMix normalizes the weighted sum of included pathway advantages by the root sum of squared weights, yielding a single *turn channel*:

$$A_{g,t}^{\mathrm{turn}} = \frac{\sum_k m_{g,t}^{(k)}\, c_k\, z_{g,t}^{(k)}}{\sqrt{\sum_k \big(m_{g,t}^{(k)}\, c_k\big)^2 + \epsilon}}. \tag{8}$$

The denominator adapts to the set of included pathways and limits scale growth as additional pathways become active. It does not guarantee unit variance: RMS-normalized pathway credits may have nonzero means and correlated fluctuations. Weights c_k specify the relative influence of normalized credit rather than the units of the raw reward.

The trajectory anchor adds trajectory-level information in parallel with the turn channel. For selected pathways, weighted, applicable immediate rewards are summed without discount over each trajectory. The resulting scalar score is standardized within the question group, with each valid trajectory counted once, to obtain the anchor B_g. The same scalar is shared across the trajectory, but contributes only at transitions that pass the gate. The two channels are fused as

$$A_{g,t} = \mathrm{clip}\Big((1-\eta) A_{g,t}^{\mathrm{turn}} + \eta\, \mathrm{gate}_{g,t} B_g,\ \pm\kappa\Big). \tag{9}$$

Here $\eta \in [0,1]$ is the anchor mixing weight, distinct from the GAE parameters λ_k, and κ is the advantage-clipping magnitude. On a failed transition, the anchor term vanishes while the turn-channel weight remains $1-\eta$. Appendix B.3 defines the anchor, and Appendix B.4 summarizes the training procedure. When the anchor is disabled, the final advantage is $\mathrm{clip}(A_{g,t}^{\mathrm{turn}}, \pm\kappa)$.

5 Experiments

5.1 Experimental setup

Main-experiment reward setting. Our search setting instantiates three common reward types with four signals. *Answer correctness* supplies terminal outcome feedback. *Retrieval redundancy* discourages revisiting the same documents, while *evidence gain* rewards newly retrieved supporting titles on eligible HotpotQA training examples; both provide local process feedback. *Format compliance* evaluates five structural requirements for search and answer actions. Process and conditional signals form the auxiliary rewards. Format is the conditional pathway: a failing turn retains only its format advantage, excluding outcome and process advantages and suppressing the trajectory anchor at that turn; a passing turn includes all applicable pathways with valid credit. These signals express task completion, productive retrieval, and valid interaction; they are representative requirements rather than an exhaustive reward taxonomy. Appendices A–F describe our reward evaluators, credit estimator, experimental protocol, hyperparameters, baseline adaptations, and search interaction interface.

Tasks and evaluation. We evaluate factual retrieval and multi-hop reasoning on seven QA benchmarks: Natural Questions (NQ) (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), PopQA (Mallen et al., 2023), HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (2Wiki) (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023). We report exact match (EM) for each benchmark and macro EM, the unweighted mean over all seven benchmarks. Evaluation uses greedy decoding with one trajectory per example.

Models and baselines. We study Qwen2.5-1.5B-Instruct, Qwen2.5-7B-Instruct, and Llama-3.1-8B-Instruct, comparing TRACER with GRPO (Shao et al., 2024), GiGPO (Feng et al., 2025), and ProxMO (Fang et al., 2026) under shared reward definitions. We also include **ProxMO-C**, a calibrated control that averages auxiliary rewards within each trajectory before adding them to the outcome score (Appendix H.2). This control tests whether simple reward-scale calibration accounts for the gains, beyond the original scalar adapters in Appendix E. A supplementary 1.5B comparison with GDPO isolates reward-wise normalization (Appendix H.1).

Training dynamics. The validation split targets 1,000 unique NQ questions and 1,000 unique HotpotQA questions. At each evaluation, validation EM is the unweighted mean of the two benchmark EM scores, giving each benchmark equal weight. Learning curves track this two-benchmark score over each method's convergence window and show both training stability and the attained plateau. This paired score differs from seven-benchmark macro EM, so baseline rankings can differ.

5.2 Main results

Table 1 gives the same macro-EM ordering on all three backbones: TRACER, ProxMO-C, ProxMO, GiGPO, and GRPO. TRACER achieves 38.81, 45.68, and 47.77 macro EM, exceeding the strongest baseline, ProxMO-C, by 1.64, 3.49, and 4.03 percentage points, respectively.

Beyond scalar reward calibration. ProxMO-C improves over ProxMO on every backbone, confirming that auxiliary-reward accumulation matters in this setting. Its remaining gap to TRACER indicates that balancing trajectory-level reward magnitudes does not recover reward-specific temporal credit, action-type comparisons, or condition-gated advantage routing. The comparison supports these mechanisms jointly; it does not imply that reward averaging eliminates all action-type bias.

5.3 Convergence and training dynamics

TRACER reaches the highest validation EM on every backbone and retains its lead at the end of the displayed windows (Figure 3b–d). ProxMO-C reaches the strongest baseline plateau across backbones, but remains below TRACER; ProxMO, GiGPO, and GRPO converge at lower levels. All three standard baselines improve in macro EM and validation plateau from 1.5B to 7B. On Llama-3.1-8B, ProxMO-C makes steady progress during early training, while the uncalibrated baselines show larger reversals; TRACER improves consistently toward a higher plateau.

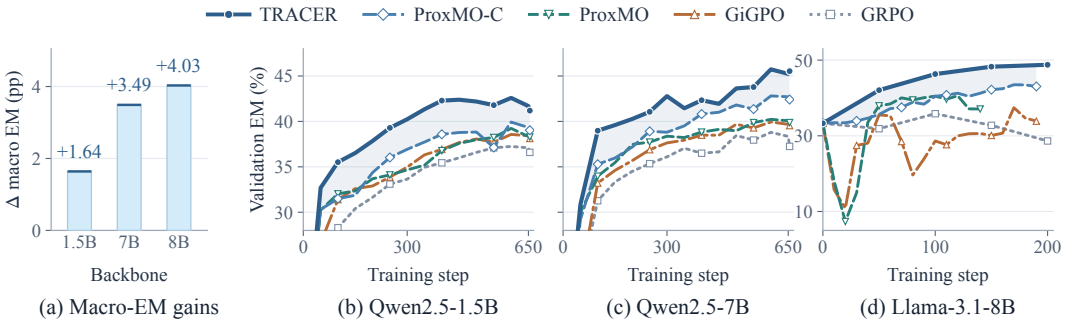
5.4 Component analysis

Table 2 examines the auxiliary-reward horizon, action-type isolation, the outcome GAE parameter λ_o, and condition gating on Qwen2.5-1.5B. It reports absolute EM and changes from full TRACER, while Figure 4(a) summarizes the benchmark profile. Full TRACER has the strongest overall performance, and the component removals produce different patterns of degradation. Each variant changes the named component while retaining the other TRACER settings, separating the effects of temporal credit, action comparability, and conditional eligibility.

Keeping auxiliary feedback local. Figure 4(d) changes the discount of all three local auxiliary pathways—format, redundancy, and evidence gain—from 0 to 0.95, retaining their separate baselines, normalization, and gate settings. The longer-horizon variant improves faster initially but levels off below full TRACER, which overtakes it later in training. Local credit keeps each auxiliary judgment attached to its current action; propagating dense auxiliary feedback across turns can make it compete with credit for task completion. The ablation thus tests their joint temporal scope.

Table 1: Test exact match (EM, %). **Bold**/underline: best/second-best per backbone. ProxMO-C uses trajectory-mean auxiliary rewards. Shading: TRACER; Δ_{best}: gain over the best baseline per column. Macro: seven-benchmark mean.

	Single-hop QA			Multi-hop QA				Overall
Method	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	**Macro**
Qwen2.5-1.5B-Instruct								
GRPO	40.39	57.84	44.09	34.10	31.23	10.63	24.80	34.73
GiGPO	41.88	57.89	42.75	35.26	33.52	11.71	28.00	35.86
ProxMO	42.02	57.92	43.65	36.45	34.22	12.78	28.80	36.55
ProxMO-C	42.30	57.78	44.28	37.56	**34.50**	11.75	32.00	37.17
TRACER (ours)	**44.29**	**59.50**	**45.20**	**40.89**	34.40	**14.56**	**32.80**	**38.81**
Δ_{best}	+1.99	+1.58	+0.92	+3.33	-0.10	+1.78	+0.80	+1.64
Qwen2.5-7B-Instruct								
GRPO	42.22	57.97	43.41	35.38	30.30	11.71	32.00	36.14
GiGPO	42.30	57.78	44.23	37.57	34.61	12.21	32.80	37.36
ProxMO	41.83	63.26	40.52	38.62	36.62	14.69	34.40	38.56
ProxMO-C	43.07	62.02	43.30	42.55	43.68	19.11	41.60	42.19
TRACER (ours)	**46.26**	**64.12**	**48.43**	**45.21**	**46.27**	**19.90**	**49.60**	**45.68**
Δ_{best}	+3.19	+0.86	+4.20	+2.66	+2.59	+0.79	+8.00	+3.49
Llama-3.1-8B-Instruct								
GRPO	38.89	54.93	42.33	32.78	29.50	10.47	32.00	34.41
GiGPO	36.98	58.55	37.36	37.87	34.81	15.76	32.00	36.19
ProxMO	39.14	61.62	43.74	42.01	45.05	19.53	40.80	41.70
ProxMO-C	43.21	65.09	46.48	43.78	42.02	18.37	47.20	43.74
TRACER (ours)	**49.75**	**66.98**	**48.65**	**47.78**	**48.31**	**22.51**	**50.40**	**47.77**
Δ_{best}	+6.54	+1.89	+2.17	+4.00	+3.26	+2.98	+3.20	+4.03



(a) Macro-EM gains (b) Qwen2.5-1.5B (c) Qwen2.5-7B (d) Llama-3.1-8B

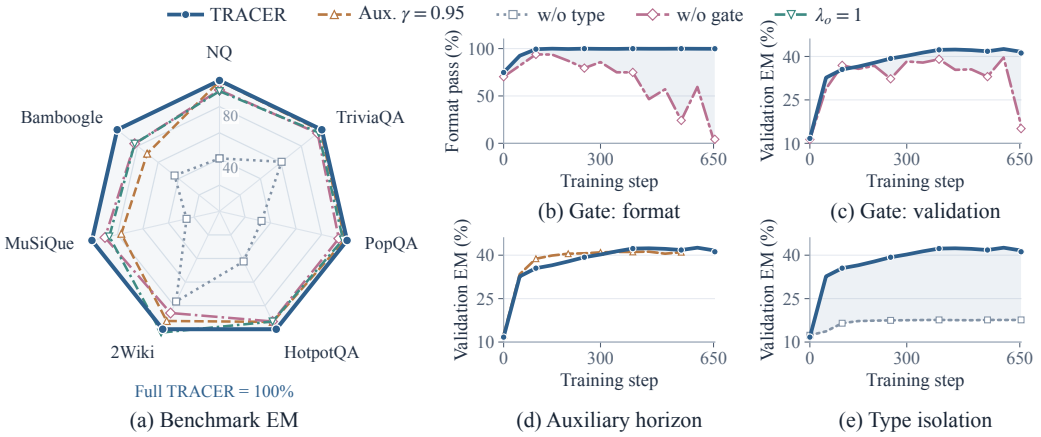
Figure 3: (a) Macro-EM gains over the strongest baseline. (b–d) NQ/HotpotQA validation EM; Qwen panels zoom to 28–48%. Shading indicates the gap between TRACER and ProxMO-C.

Comparing like actions. Without action-type isolation, validation gains are slow and plateau well below full TRACER (Figure 4e), alongside broad degradation in Table 2. Mixed pools let differences in search and answer return distributions bias their relative credit. Action-type isolation compares each search or answer decision with decisions of the same type, reducing the influence of between-type return differences on its baseline and normalization scale. Appendix G.1 shows persistently lower mean advantages for search actions and collapse toward single-turn answering with almost no retrieval, consistent with this loss of performance.

GAE hyperparameter sensitivity. The GAE parameter λ_k weights multi-step TD residuals (Schulman et al., 2016). Increasing the outcome GAE parameter λ_o from 0.95 to 1 at fixed $\gamma_o = 0.95$ reduces macro EM by 2.33 percentage points (Table 2). The default thus outperforms the discounted Monte Carlo limit here. Local pathways are unaffected: $\gamma = 0$ makes their credit independent of λ.

Table 2: Qwen2.5-1.5B ablations: test EM (%) above, signed change from full TRACER below (pp). Macro: seven-benchmark mean; **bold**: column best. Validation curves use NQ/HotpotQA.

	Single-hop QA			Multi-hop QA				Overall
Variant	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	**Macro**
TRACER (full)	**44.29** (0.00)	**59.50** (0.00)	**45.20** (0.00)	**40.89** (0.00)	34.40 (0.00)	**14.56** (0.00)	**32.80** (0.00)	**38.81** (0.00)
Aux. $\gamma = 0.95$	44.10 (-0.19)	58.84 (-0.66)	44.41 (-0.79)	38.35 (-2.54)	32.00 (-2.40)	11.21 (-3.35)	23.20 (-9.60)	36.02 (-2.79)
w/o action-type isolation	17.92 (-26.37)	36.12 (-23.38)	14.89 (-30.31)	17.39 (-23.50)	26.31 (-8.09)	3.76 (-10.80)	14.40 (-18.40)	18.68 (-20.13)
w/o condition gate	41.08 (-3.21)	57.02 (-2.48)	42.26 (-2.94)	38.20 (-2.69)	29.67 (-4.73)	13.07 (-1.49)	27.20 (-5.60)	35.50 (-3.31)
GAE $\lambda_o = 1$	40.66 (-3.63)	57.54 (-1.96)	43.77 (-1.43)	38.22 (-2.67)	**35.39** (+0.99)	12.58 (-1.98)	27.20 (-5.60)	36.48 (-2.33)



(a) Benchmark EM (b) Gate: format (c) Gate: validation (d) Auxiliary horizon (e) Type isolation

Figure 4: Qwen2.5-1.5B ablations. (a) Benchmark EM relative to full TRACER (100%; Table 2). (b–c) Gate: trajectory format pass rate and validation EM. (d) Auxiliary horizon: $\gamma = 0$ versus 0.95 for format and both process signals. (e) Action-type isolation (type). EM uses NQ/HotpotQA from initialization; curves end at their last evaluation. Curve fills mark TRACER's gains.

Condition gating and reliable interaction. Disabling condition gating retains the format reward but allows outcome and process credit, as well as the trajectory anchor, to contribute on format-failing turns, reducing macro EM by 3.31 percentage points (Table 2). In Figure 4(b–c), full TRACER rapidly learns the required format and maintains high compliance alongside strong validation performance. Without the gate, lower compliance deteriorates later as validation EM drops sharply. In practical tool use, reliable learning must preserve both answer quality and the interaction protocol that allows the agent to execute its actions reliably throughout training.

6 Conclusion

Practical multi-turn agents face diverse reward requirements, making credit assignment a joint problem of temporal scope, action comparability, and conditional eligibility. TRACER addresses this problem by estimating reward-specific advantages, comparing semantically compatible actions, and routing credit according to task-defined prerequisites before fusion. Experiments on search-augmented question answering across seven benchmarks and three backbones show improved answer accuracy, while component analyses connect these gains to local auxiliary credit, sustained search behavior, and format compliance. These findings support preserving reward semantics throughout credit estimation in practical agent training. Extending this evaluation to other tool-use domains and richer reward dependencies is a natural next step.

AI USE STATEMENT

We used generative AI tools to assist with English-language polishing of the manuscript. This assistance included correcting grammar and punctuation, refining word choice and sentence structure, and improving clarity, concision, and fluency. We also used these tools to smooth transitions between sentences and paragraphs, reduce repetitive phrasing, and maintain consistent terminology and academic tone throughout the paper.

ETHICS STATEMENT

This work studies credit assignment in a bounded, text-based retrieval environment using existing question-answering benchmarks. Its goal is to support reliable learning from task, process, and interaction requirements. Condition-gated routing applies specified prerequisites rather than providing a general guarantee of factual accuracy or safety. Broader use should account for corpus biases and outdated information through domain-specific evaluation, appropriate data governance, and monitoring of answer quality and tool-use behavior.

REPRODUCIBILITY STATEMENT

The appendix documents reward definitions, applicability masks, and condition-gated routing (Appendix A), together with the credit estimator and training procedure (Appendix B). Appendix C describes the model and data choices, validation-split construction, retrieval environment, interaction limits, and evaluation and checkpoint-selection protocols. Shared optimization settings and pathway-specific hyperparameters are provided in Appendix D; matched baseline adaptations and ablation controls are specified in Appendix E. The action interface and supplementary training diagnostics appear in Appendices F and G. Additional multi-reward baseline comparisons and the calibrated scalar-reward control are detailed in Appendix H. We will open-source the code repository after the review process for this paper concludes.

REFERENCES

Jie Cheng, Gang Xiong, Ruixi Qiao, Lijun Li, Chao Guo, Junle Wang, Yisheng Lv, and Fei-Yue Wang. Stop Summation: Min-Form Credit Assignment Is All Process Reward Model Needs for Reasoning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 131646–131671, 2025. doi: 10.52202/085713-4382.

Yangyi Fang, Jiaye Lin, Xiaoliang Fu, Cong Qin, Chang Liu, and Haolin Shi. Proximity-Based Multi-Turn Optimization: Practical Credit Assignment for LLM Agent Training. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 285–307, 2026. doi: 10.18653/v1/2026.acl-industry.19.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-Group Policy Optimization for LLM Agent Training. In *Advances in Neural Information Processing Systems*, volume 38, pp. 46375–46408, 2025. doi: 10.52202/085713-1544.

Shuo He, Lang Feng, Qi Wei, Xin Cheng, Lei Feng, and Bo An. Hierarchy-of-Groups Policy Optimization for Long-Horizon Agentic Tasks. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=T8Dev99qnz.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020. doi: 10.18653/v1/2020.coling-main.580.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O. Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Rwhi91ideu.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, 2017. doi: 10.18653/v1/P17-1147.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. In *International Conference on Learning Representations*, pp. 39578–39601, 2024. URL https://arxiv.org/abs/2305.20050.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. GDPO: Group reward-Decoupled Normalization Policy Optimization for Multi-reward RL Optimization. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2601.05242.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, 2023. doi: 10.18653/v1/2023.acl-long.546.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and Narrowing the Compositionality Gap in Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, 2023. doi: 10.18653/v1/2023.findings-emnlp.378.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tur, Gokhan Tur, and Heng Ji. ToolRL: Reward is All Tool Learning Needs. In *Advances in Neural Information Processing Systems*, volume 38, pp. 105523–105553, 2025. doi: 10.52202/085713-3524.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In *International*

Conference on Learning Representations, 2016. URL https://arxiv.org/abs/1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multi-hop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475.

Guoqing Wang, Sunhao Dai, Guangze Ye, Zeyu Gan, Wei Yao, Yong Deng, Xiaofeng Wu, and Zhenzhe Ying. Information Gain-based Policy Optimization: A Simple and Effective Approach for Multi-Turn Search Agents. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=qkWP6phrvZ.

Quan Wei, Siliang Zeng, Chenliang Li, Zhongruo Wang, William Brown, Oana Frunza, Wei Deng, Anderson Schneider, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, and Mingyi Hong. Reinforcing Multi-Turn Reasoning in LLM Agents via Fine-Grained Reward Structure and Credit Assignment. *arXiv preprint arXiv:2505.11821v3*, 2026. URL https://arxiv.org/abs/2505.11821v3.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-Grained Human Feedback Gives Better Rewards for Language Model Training. In *Advances in Neural Information Processing Systems*, volume 36, pp. 59008–59033, 2023. doi: 10.52202/075280-2574.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018. doi: 10.18653/v1/D18-1259.

Yimeng Zhang, Tian Wang, Jiri Gesi, Ziyi Wang, Yuxuan Lu, Jiacheng Lin, Sinong Zhan, Vianne Gao, Ruochen Jiao, Junze Liu, Kun Qian, Yuxin Tang, Ran Xue, Houyu Zhang, Qingjun Cui, Yufan Guo, and Dakuo Wang. Shop-R1: Rewarding LLMs to Simulate Human Behavior in Online Shopping via Reinforcement Learning. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2507.17842v2.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. DeepResearcher: Scaling Deep Research via Reinforcement Learning in Real-world Environments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 414–431, 2025. doi: 10.18653/v1/2025.emnlp-main.22.

A REWARD DESIGN

Our search implementation instantiates three reward types with four signals: answer correctness (outcome), retrieval redundancy and evidence gain (process), and format compliance (conditional). All four are computed from reference answers, structural checks, or retrieval records. Let g index a trajectory for question q, $t = 0, \ldots, T_g - 1$ index its actions, and $k \in \{o, f, d, e\}$ index the four signals. We denote reward applicability by $b_{g,t}^{(k)} \in \{0, 1\}$, distinguishing an observed zero score from an unavailable or inapplicable reward.

A.1 OUTCOME: TERMINAL ANSWER CORRECTNESS

The outcome reward is binary exact match (EM), with $\mathbb{I}$ denoting an indicator:

$$r_{g,t}^{(o)} = \mathbb{I}[t = T_g - 1] \max_{y \in Y_q} \mathbb{I}[\mathrm{norm}(\hat{y}_g) = \mathrm{norm}(y)], \tag{10}$$

where Y_q contains the reference answers. Normalization lowercases text, removes English punctuation and the articles `a/an/the`, and collapses whitespace. The predicted answer is extracted from the last complete `<answer>...</answer>` block in the environment's accumulated history, after action projection; a missing extractable answer receives zero. The immediate reward occurs only at termination, but outcome credit remains applicable to earlier actions so that successful answers can reinforce preceding searches. The matched search configuration disables invalid-action reward penalties and reward-side KL terms, leaving this signal as pure answer correctness. Reference-policy KL regularization is applied separately in the actor loss (Appendix D.1).

A.2 CONDITIONAL REWARD: FORMAT COMPLIANCE

Format is scored on each turn's *raw model output*, before action projection. The evaluator averages five binary checks:

$$r_{g,t}^{(f)} = \frac{1}{5} \sum_{j=1}^{5} \mathbb{I}[\text{check } j \text{ passes}]. \tag{11}$$

The expected action is search on a nonterminal turn and answer on a terminal turn. Table 3 specifies the checks; the passing threshold 0.95 requires all five to pass because scores lie in $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$. This is a structural evaluator, not a test of reasoning quality, query relevance, or answer correctness.

Table 3: Per-turn format checks; action blocks must match the expected action type.

Check	Passing rule
Think block	Exactly one opening `<think>` tag and a complete think block; its content need not be nonempty.
Action block	A complete search or answer block of the expected type, with nonempty content after trimming whitespace.
Single action type	A complete action block and no simultaneous search and answer opening tags.
Single action instance	A complete action block, with at most one opening tag of each action type.
No trailing text	Only whitespace follows the first complete action block.

Action projection and format scoring serve different purposes and need not agree on whether an action is acceptable. An executable answer can be correct but omit the think block, yielding at most 0.8 format reward. Conversely, a search at the final action limit remains a search-type action, but fails the terminal answer-block requirement even if its search tags are complete. The checks do not constitute a full XML grammar validator.

A.3 PROCESS REWARD: RETRIEVAL REDUNDANCY

Let $D_{g,t}$ be the list of documents parsed from the current retrieval observation and $U_{g,<t}$ the set of documents previously observed in the trajectory. For a nonempty successful retrieval, the redundancy reward is

$$r_{g,t}^{(d)} = -\frac{\sum_{x \in D_{g,t}} \mathbb{I}[x \in U_{g,<t}]}{|D_{g,t}|}. \tag{12}$$

It ranges from 0 for entirely new documents to -1 for entirely repeated documents. Document identity is based on whitespace-normalized body text parsed from the observation's information JSON and document markers, rather than query similarity or a corpus identifier. The seen set is updated *after* scoring: duplicates within the current retrieval are not treated as repetitions of an earlier search, although each list entry remains in the denominator. The first valid retrieval therefore has score zero. Answer turns, failed retrievals, and observations with no parsed documents are inactive, with a numerical placeholder of zero; this placeholder does not certify novelty.

A.4 PROCESS REWARD: EVIDENCE GAIN

Let H_q denote the set of gold supporting titles, $R_{g,t}$ the titles retrieved at the current turn, and $S_{g,<t}$ the gold titles already retrieved. Evidence gain is the fraction of previously unseen gold titles found by the current search:

$$r_{g,t}^{(e)} = \frac{|(R_{g,t} \cap H_q) \setminus S_{g,<t}|}{|H_q|}. \tag{13}$$

Gold titles are obtained from HotpotQA supporting-fact annotations. A training example is eligible only when its gold-title set is nonempty and every title can be located in the retrieval corpus; other examples, including NQ examples without this supervision, remain in training with evidence-gain reward inactive. On an eligible example, a completed search with status `success` or `no_results` produces an observable score, whereas a tool error makes the reward inactive. A supporting title earns evidence credit only once, so the total raw evidence-gain reward per trajectory is at most one.

Title normalization applies HTML and URL decoding, converts underscores to spaces, performs Unicode NFKC normalization, collapses whitespace, strips surrounding double quotation marks, and case-folds. The document title field is preferred; otherwise the first line of its contents is used. Gold titles are available only to the reward evaluator and are not inserted into the policy observation. This signal measures title-level supporting-evidence coverage, not sentence-level entailment or complete reasoning correctness.

A.5 APPLICABILITY AND CONDITION-GATED ROUTING

Condition-gated routing operates at the advantage-fusion layer. Let $e_{g,t}^{(k)}$ indicate that pathway k is applicable and has a valid estimated advantage on a nontruncated transition. Using the pass decisions in Equation 7, the inclusion mask is

$$m_{g,t}^{(k)} = e_{g,t}^{(k)} \begin{cases} 1, & \text{gate}_{g,t} = 1, \\ \mathbb{I}[k \in \mathcal{C} \ \wedge \ \neg \text{pass}_{g,t}^{(k)}], & \text{gate}_{g,t} = 0. \end{cases} \tag{14}$$

An inapplicable condition is treated as passed but contributes no advantage of its own. When multiple conditions are configured, failure retains only the failed conditions with valid credit; conditions that already pass are excluded along with all nonconditional pathways. If no condition is configured, the gate is open on every valid transition.

In the four-signal search setting, format is the sole conditional pathway, with threshold 0.95. A passing turn includes every applicable pathway with valid credit. A failing turn includes only format credit: outcome, redundancy, and evidence-gain advantages are excluded from both the RSSMix numerator and its weight norm. The entire trajectory anchor is also zeroed at that turn; other turns retain their own gate decisions. Disabling the condition gate keeps format as an ordinary reward pathway and includes all valid, applicable credits regardless of its score.

Gate decisions use immediate conditional rewards, but do not alter the reward inputs to returns, baselines, or GAE. Normalization is computed before selection, so a turn excluded from fusion by

a gate may still contribute to its pathway's comparison statistics. Applicability, credit validity, and conditional inclusion are separate decisions; an included failed-condition advantage can have either sign.

B Implementation of Credit Estimation

This section specifies the history-aware estimator and its edge cases. A turn is one model action and its environment interaction; a training step is one rollout-and-update iteration. Rows replicated for data-parallel divisibility are deduplicated by trajectory identifier and turn index before credit statistics are computed, then assigned the corresponding restored advantages. They do not count as additional sampled trajectories.

B.1 Comparison pools and history-aware values

Within each question, candidate pools are refined by action type and, for selected pathways, turn index. Action type is determined from the projected action: a complete answer is assigned to the answer pool and other actions to search. The reference configuration uses leave-one-trajectory-out comparisons: every transition from the target trajectory is excluded from value estimation. For $x = (g, t)$, the eligible candidate set satisfies $\mathcal{C}_x^{(k)} \subseteq \{(g', t') \in \mathcal{P}_x^{(k)} : g' \neq g\}$, and

$$w_{xy}^{(k)} = \frac{\exp(K_{xy}/\tau_{\text{sim}})}{\sum_{z \in \mathcal{C}_x^{(k)}} \exp(K_{xz}/\tau_{\text{sim}})}, \qquad V_x^{(k)} = \sum_{y \in \mathcal{C}_x^{(k)}} w_{xy}^{(k)} G_y^{(k)}, \tag{15}$$

where $G^{(k)}$ is the discounted return of the pathway's immediate reward and $\tau_{\text{sim}} = 0.1$.

The similarity kernel uses action–retrieval-result pairs strictly *before* the current action. Each history window is whitespace-normalized, boundary-padded, and represented by hashed character 2–4-gram counts using `crc32` modulo 2^{18}, followed by L2 normalization. Window text is capped at 8,000 characters; the environment exposes at most eight windows and the reference estimator uses the most recent three. Writing $\psi(h_{x,-j})$ for window j, we form

$$u_x = \text{L2Norm}\left(\sum_{j=1}^{\min(W,t)} \rho^j \psi(h_{x,-j})\right), \qquad z_x = \text{L2Norm}(u_x \odot \text{idf}), \qquad K_{xy} = z_x^\top z_y. \tag{16}$$

Here $W = 3$, $\rho = 1$, and $\text{idf}(v) = \log((1+n)/(1+\text{df}(v))) + 1$ is fitted within the question–action-type group using only histories from other trajectories. This peer-fitted IDF transforms both target and candidate histories. The first action has an empty history vector and hence uniform candidate weights. The e5 encoder used by the retriever is not used for this state representation.

If turn-index refinement leaves no candidates, ordinary pathways may relax turn-index matching while retaining question, action type, and trajectory exclusion; normalization still uses the original pool. Evidence gain does not allow fallback across turn indices and uses only active rows for its value candidates. Other pathways can retain inactive rows with zero immediate reward in return/value calculations, then apply the active mask during normalization and fusion. An empty candidate pool is marked missing; no transition from the target trajectory is reintroduced. The ablation without action-type isolation removes action-type refinement of comparison and normalization pools; the IDF-fitting scope remains unchanged.

B.2 Returns, GAE, and normalization

For immediate rewards $r^{(k)}$, credit is computed independently along each trajectory, before condition-gated routing:

$$G_{g,t}^{(k)} = r_{g,t}^{(k)} + \gamma_k G_{g,t+1}^{(k)}, \tag{17}$$

$$\delta_{g,t}^{(k)} = r_{g,t}^{(k)} + \gamma_k V_{g,t+1}^{(k)} - V_{g,t}^{(k)}, \qquad a_{g,t}^{(k)} = \delta_{g,t}^{(k)} + \gamma_k \lambda_k a_{g,t+1}^{(k)}. \tag{18}$$

Post-termination returns, values, and advantages are zero. Missing adjacent successors or required values interrupt the corresponding credit recursion. An inactive terminal evidence-gain state with

zero reward has a known zero value; unknown intermediate evidence-gain values can still interrupt a nonlocal evidence-gain GAE recursion.

The reference configuration uses RMS normalization over valid, active turns in the pathway's comparison pool $\mathcal{P}_{g,t}^{(k)}$:

$$z_{g,t}^{(k)} = \frac{a_{g,t}^{(k)}}{\sqrt{\mathrm{mean}_{x \in \mathcal{P}_{g,t}^{(k)}}[(a_x^{(k)})^2]} + 10^{-6}}. \tag{19}$$

This preserves the sign of the raw credit and does not impose a zero mean. The implementation also supports centered sample-standard-deviation normalization and division by sample standard deviation without centering. Standard-deviation modes zero singleton pools; RMS can preserve a singleton's sign if its value and credit are valid. Zero or nonfinite scales produce zero output.

Limiting cases. At $\gamma_k = 0$, the return and raw credit are local, with $a^{(k)} = r^{(k)} - V^{(k)}$, independently of λ_k. On a complete trajectory with available values, setting $\lambda_k = 1$ telescopes the TD residuals to $G^{(k)} - V^{(k)}$ for the chosen discount. These identities describe the raw pathway credit; normalization, routing, and trajectory fusion can still couple the final advantages.

B.3 Fusion and the routed trajectory anchor

RSSMix divides the weighted sum of included, normalized pathway advantages by the L2 norm of their included weights, as in Equation 8. A valid included pathway with zero advantage still contributes its nonzero weight to that denominator; an inactive or gated-off pathway does not. The denominator controls scale growth but does not guarantee unit variance, since pathways may be correlated and RMS-normalized credit need not be centered.

The reference anchor uses outcome and evidence gain, excluding format and redundancy. For each trajectory, let

$$S_g^{(k)} = c_k \sum_t b_{g,t}^{(k)} r_{g,t}^{(k)}, \qquad S_g = \sum_{k \in \{o,e\}} S_g^{(k)}. \tag{20}$$

These are undiscounted immediate-reward sums, not initial return-to-go values. Each valid trajectory contributes once to anchor statistics. The total score is standardized within each question group, then the whole anchor is gated separately at each transition:

$$\begin{aligned} B_g &= \frac{S_g - \mathrm{mean}_{j \in q}\, S_j}{\mathrm{std}_{j \in q}(S_j) + 10^{-6}}, \\ A_{g,t} &= \mathrm{clip}(0.9 A_{g,t}^{\mathrm{turn}} + 0.1\, \mathrm{gate}_{g,t} B_g, -5, 5). \end{aligned} \tag{21}$$

The standard deviation uses one degree-of-freedom correction. The scalar B_g is shared across turns; only its inclusion varies with the per-transition gate. A failing turn receives zero anchor contribution while retaining the turn-channel weight 0.9; passing turns in the same trajectory are unaffected by that gate decision. Groups with fewer than two valid trajectories or no score variation receive zero anchor. When the anchor is disabled, the turn channel retains weight one.

B.4 TRAINING PROCEDURE

Algorithm 1: TRACER for multi-reward search

Inputs: policy, question batch, trajectories per question, reward evaluators, pathway scopes, conditional pathways and thresholds.

1. Sample grouped multi-turn trajectories; record raw/projected actions, histories, rewards, active masks, and termination flags.
2. Deduplicate transitions and identify valid, applicable rewards for each pathway.
3. For each pathway, compute discounted returns, history-based leave-one-trajectory-out values, and GAE advantages, respecting missing-value boundaries.
4. Normalize valid active credits within pathway pools. Evaluate conditions on immediate rewards; retain all eligible pathways on passing turns and only failed-condition pathways otherwise. Combine selected advantages with RSSMix.
5. Standardize the trajectory score from outcome and evidence-gain rewards. Mask the entire anchor at failed turns, then fuse and clip the turn-level advantages.
6. Restore duplicated rows, broadcast each turn's scalar advantage to its response tokens, and update the policy with clipped optimization and reference-policy KL regularization.

C EXPERIMENTAL PROTOCOL

C.1 MODELS AND QUESTION-ANSWERING DATA

We use Qwen2.5-1.5B-Instruct, Qwen2.5-7B-Instruct, and Llama-3.1-8B-Instruct; shortened names omit the Instruct suffix. Training uses the NQ–HotpotQA mixture from PeterJinGo's `nq_hotpotqa_train` release. Evaluation covers NQ, TriviaQA, PopQA, HotpotQA, 2WikiMultiHopQA, MuSiQue, and Bamboogle, with references in §5.1. The first three primarily evaluate factual retrieval, while the remaining benchmarks probe multi-hop or compositional search. The matched GRPO, GiGPO, ProxMO, ProxMO-C, and TRACER configurations share the search environment, reward observations, model backbones, and data protocol; the GDPO comparison uses the same 1.5B setting.

Independent validation (development) split. The preprocessing pipeline normalizes question keys, removes training questions whose keys overlap the seven evaluation sources, and creates a validation split from the remaining training data. Within each source, unique question keys are ordered by a SHA-256 hash of the split seed and question key; split seed 42 selects 1,000 unique questions each from NQ and HotpotQA. All records sharing a selected key are moved together. Thus the target is 2,000 unique validation questions, while the physical number of records can differ when duplicate records exist. Question-key exclusion is distinct from a semantic contamination audit.

C.2 SEARCH ENVIRONMENT AND INTERACTION LIMITS

Retrieval uses an e5-base-v2 encoder and a FAISS index over an offline Wikipedia-2018 corpus, returning three documents per query. The seven evaluation benchmarks use the same retrieval service. The maximum trajectory length is six model actions, including the final answer; a sixth action that continues searching terminates before another tool call, so at most five searches are executed. Termination by budget exhaustion is not relabeled as an answer action.

The policy observation can retain six turns of interaction history. Each model call allows up to 8,192 prompt tokens and 512 response tokens, with left truncation of an overlong prompt; these are per-turn rather than whole-trajectory limits. The rollout engine's model-length limit is 8,704 tokens. Its batched-token budget, 16,384, is a scheduling limit rather than additional context capacity. The credit kernel uses three preceding action–retrieval windows (Appendix B.1).

C.3 Evaluation and checkpoint selection

Evaluation uses one greedy trajectory per question, with temperature zero and sampling disabled. An answer is successful when the trajectory submits an answer and its terminal outcome reward is positive; format failure is not an additional filter on EM. Seven-benchmark macro EM is the unweighted mean of benchmark EM scores. The trajectory-weighted overall EM is a different statistic and is not substituted for macro EM. Validation curves use the mean of NQ and HotpotQA EM, so their ordering can differ from seven-benchmark macro EM.

The current four-signal command configuration evaluates before training and at training completion. Intermediate validation is scheduled every 10 training steps for GRPO, GiGPO, ProxMO, ProxMO-C, and GDPO, and every 50 training steps for TRACER and the outcome-only configuration. The five checkpoints with the highest validation EM are retained for subsequent full-benchmark evaluation. The reporting rule selects the checkpoint by validation EM and uses its complete seven-benchmark test vector; different benchmarks must not select different checkpoints. The differing validation cadence gives methods different candidate densities and should be retained when interpreting this protocol; a common-step analysis would be a separate control.

C.4 Compute and run-level reproducibility

The launch configuration allocates eight GPUs per run, uses FSDP and bfloat16 generation, and enables gradient checkpointing and padding removal. Actor parameters and optimizer states are not offloaded; reference-model parameters are offloaded.

Single-seed evaluation. We use one training run per reported configuration with seed 0 for the environment, data ordering, and rollout engine. This resource constraint reflects the cost of multi-turn search training, including repeated model generation and retrieval within each trajectory. Scores describe individual runs rather than averages over independent seeds; successive checkpoints, ablation variants, and archived validation runs are not additional repeats of the same configuration. We do not report across-seed standard deviations or claim that the observed rankings are statistically significant. Curve fluctuations characterize within-run dynamics rather than uncertainty across training runs. The fixed seed does not guarantee token-level determinism across distributed executions. Training uses one epoch over the filtered training set, dropping incomplete final batches; the update count depends on the data and loader.

D Hyperparameters

The tables separate shared optimization settings, model-dependent resource settings, and TRACER-specific credit controls. They specify the four-signal reference configuration used throughout the method and experiment descriptions.

D.1 Shared policy optimization and sampling

A training batch contains 256 questions with five sampled trajectories per question, producing 1,280 trajectories. These expand into variable-length turn records; the PPO minibatch size therefore counts expanded records rather than original questions. Each turn's scalar advantage is broadcast to its valid response tokens. With ratio $\varrho = \pi_\theta/\pi_{\text{old}}$, the actor first computes

$$\ell_{\text{clip}} = \max[-\varrho A, -\operatorname{clip}(\varrho, 0.8, 1.2)A], \qquad \ell = \begin{cases} \min(\ell_{\text{clip}}, -3A), & A < 0, \\ \ell_{\text{clip}}, & A \geq 0. \end{cases} \tag{22}$$

Losses are averaged over valid response tokens and augmented by reference-policy KL with coefficient 0.001. The low-variance KL estimator is $\exp(z) - z - 1$ for $z = \log \pi_{\text{ref}} - \log \pi_\theta$; its output is clipped to $[-10, 10]$. This actor regularizer is separate from rewards.

D.2 Pathway and fusion settings

Table 6 gives the four-signal configuration. The three auxiliary pathways have $\gamma = 0$, making their configured GAE λ irrelevant to raw temporal credit. The outcome pathway uses $\gamma_o = \lambda_o = 0.95$, so its recursive trace multiplier is 0.9025. All use RMS scaling and leave-one-trajectory-out values.

Table 4: Shared optimization, sampling, and environment settings.

Parameter	Value
Optimization	
Questions per training batch	256
Trajectories per question	5
Training budget	1 epoch
PPO update epochs	1
Optimizer	AdamW
Learning rate	10^{-6}
Adam β_1	0.9
Adam β_2	0.999
Weight decay	0.01
Learning-rate warmup fraction	10% of steps
Post-warmup learning-rate schedule	Constant
PPO minibatch size	512 turn records
Gradient norm clip	1.0
PPO ratio clipping interval	$[0.8, 1.2]$
Dual-clip coefficient	3
Actor KL coefficient	0.001
Entropy coefficient	0
Invalid-action reward penalty	Disabled
Reward-side KL	Disabled
Sampling and rollout resources	
Training temperature	1.0
Training top-p	1.0
Training top-k	-1 (disabled)
Evaluation trajectories per question	1
Evaluation decoding	Greedy
Rollout model-length limit	8,704 tokens
Rollout batched-token budget	16,384 tokens
Rollout memory utilization fraction	0.6
Environment and history	
Maximum model actions per trajectory	6
Maximum executed searches per trajectory	5
Prompt limit per turn	8,192 tokens
Response limit per turn	512 tokens
Policy history limit	6 turns
Credit-estimator history limit	3 windows
Retrieval encoder	e5-base-v2
Documents returned per search	3

Table 5: Model-specific settings. TP: rollout tensor parallelism; microbatches are per GPU.

Backbone	GPUs	TP	Actor microbatch	Log-prob. microbatch
Qwen2.5-1.5B-Instruct	8	1	4	8
Qwen2.5-7B-Instruct	8	2	1	2
Llama-3.1-8B-Instruct	8	2	1	2

Table 6: TRACER pathways. Type/turn index refine pools; only format is conditional.

Pathway	Weight	Discount γ	GAE λ	Threshold	Type	Turn
Outcome	1.0	0.95	0.95	—	Yes	No
Format	0.625	0	0.95	0.95	Yes	No
Redundancy	0.0625	0	0.95	—	Yes	Yes
Evidence gain	0.3125	0	0.95	—	Yes	Yes

The state kernel uses character TF-IDF, three history windows, history weight $\rho = 1$, and softmax temperature 0.1. Condition-gated routing selects normalized advantages at each turn: failed format retains only format credit and disables that turn's anchor contribution. The anchor includes outcome and evidence gain; its mixing weight is $\eta = 0.1$ and the final advantage is clipped to $[-5, 5]$. Numerical stabilization uses $\epsilon = 10^{-6}$, added after the RMS square root in the normalization denominator and inside the square root of the RSSMix weight norm as shown in the corresponding formulas. The coefficient budget is $1 : (0.625 + 0.0625 + 0.3125) = 1 : 1$ for outcome versus auxiliary signals; this does not imply equal realized rewards, advantages, or gradient contributions.

D.3 Additional reward configurations

Alongside the four-signal setting, we specify an outcome-only setting and a setting without format reward. Table 7 summarizes their reward coefficients and routing rules. Removing format reallocates the process coefficient budget and disables routing, so it is not a single-factor gate ablation. The outcome-only setting retains TRACER's estimator and an outcome anchor; it does not reduce to the GRPO implementation.

Table 7: Reward coefficients ($c_o = 1$ by default; optional $c_o = 2, 3$).

Configuration	Outcome	Format	Redundancy	Evidence gain	Routing
Four signals	c_o	0.625	0.0625	0.3125	Condition gate
No format (B2)	c_o	0	$1/6$	$5/6$	Disabled
Outcome only (B0)	1	0	0	0	Disabled

These configurations distinguish the reward requirements supplied to the estimator. The main comparisons use the four-signal setting, while the gate ablation retains all four signals and changes only their condition-gated routing.

E Baseline Adaptations and Component Controls

E.1 Matched reward inputs

We distinguish the original GRPO, GiGPO, and ProxMO estimators from their adaptations to the present multi-reward search setting. The matched baseline adapter uses the same immediate reward definitions and coefficients as TRACER, with estimator-specific routing as detailed below. The baseline adapter applies reward applicability and a search-format check to process rewards before scalarization:

$$r_{g,t} = c_o r^{(o)}_{g,t} + c_f r^{(f)}_{g,t} + \sum_{k\in\{d,e\}} c_k b^{(k)}_{g,t} F_{g,t} r^{(k)}_{g,t}, \qquad G_{g,t} = r_{g,t} + 0.95 G_{g,t+1}. \tag{23}$$

Here $F_{g,t} = \mathbb{I}[r_{g,t}^{(f)} \geq 0.95]$ is the baseline adapter's reward-side format check; it is distinct from TRACER's advantage-level inclusion mask. The trajectory score is $G_{g,0}$, not the sum of suffix returns. Format remains its positive score; it is neither shifted by minus one nor moved to termination. Thus all reward components share the same temporal discount in these baseline adapters. ProxMO-C changes this scalarization by replacing auxiliary sums with trajectory means; its definition and comparison purpose appear in Appendix H.2.

E.2 Trajectory and step estimators

GRPO. For each question, initial returns are standardized across trajectories using sample standard deviation and stabilizer 10^{-6}. The resulting advantage is broadcast to every turn of its trajectory.

GiGPO. The trajectory term is the standardized initial return. The step term standardizes suffix returns within discrete anchor-observation groups, and the two terms are added with step weight 1.0. The configured grouping compares an observation with each cluster representative using `SequenceMatcher` similarity and inserts it into the first cluster meeting threshold 0.9. Local groups include the current transition, so singleton centering yields zero step credit. This adapter uses neither TRACER's explicit action-type/turn-index pools nor its history kernel. These matched-adapter settings do not imply that all GiGPO variants use this grouping rule (Feng et al., 2025).

ProxMO. ProxMO constructs soft baselines using TF-IDF cosine similarity and temperature-scaled weights (Fang et al., 2026). Our adapter fits word TF-IDF with at most 1,024 features to batch anchor observations and computes weighted values within each question at temperature 0.1. It includes the current transition, performs no explicit action-type refinement, and subtracts the value from suffix return without step standard-deviation normalization. An empty TF-IDF vocabulary gives zero step credit. The trajectory term centers initial returns and applies success-rate modulation based on *binary answer success*, not on the potentially nonbinary mixed return. With empirical success rate p, the weights are

$$w_+ = 1+\beta[\sigma((1-p)^\alpha)-0.5], \qquad w_- = 1+\beta[\sigma(-p^\alpha)-0.5], \quad \alpha = 4, \quad \beta = \frac{0.05}{\sigma(1)-0.5}. \tag{24}$$

Here σ is the sigmoid and $\beta \approx 0.2164$ limits the maximum deviation from one to approximately 0.05. The final advantage adds this trajectory term to the step term with weight 1.0. The modulation affects only the trajectory term.

E.3 Controlled changes

The component configurations in Table 8 change specific parts of the reference estimator. In particular, disabling the gate retains format reward, whereas removing format changes both the reward collection and its coefficient allocation. The variant without action-type isolation retains turn-index refinements and the action-type-specific scope used to fit IDF. Changing the GAE hyperparameter λ_o from 0.95 to 1 leaves the outcome discount γ_o at 0.95.

Table 8: Ablation controls relative to the four-signal setting.

Control	Configuration change
Auxiliary horizon	Set format, redundancy, and evidence-gain discounts from 0 to 0.95.
w/o action-type isolation	Disable action-type refinement for pathway values and normalization; keep turn-index settings and IDF fitting unchanged.
w/o condition gate	Disable format's conditional role: include all valid, applicable pathway advantages and retain the anchor on format-failing turns; keep format feedback and coefficients.
GAE $\lambda_o = 1$	Change only the outcome pathway's GAE λ_o from 0.95 to 1.
Shorter interaction	Reduce the action limit from six to four; this changes the interaction budget, not just credit estimation.

E.4 Reward accumulation and diagnostic interpretation

A simple calculation illustrates why coefficient balance does not fix temporal balance. For a six-action trajectory with full format score at every turn, the baseline format contribution to the initial return is $0.625 \sum_{t=0}^{5} 0.95^t \approx 3.311$, whereas a correct terminal outcome contributes $0.95^5 \approx 0.774$. These are reward-level quantities before baseline subtraction and normalization; they do not determine a gradient ratio or prove that an estimator must fail. A local format discount removes this particular accumulation, while comparison statistics and fusion further determine policy credit. Trajectory averaging also removes the growth of bounded auxiliary scores with the number of scored turns, which motivates ProxMO-C as a separate control. Consequently, the uncalibrated adapters alone cannot distinguish the benefit of reward rescaling from that of a different credit estimator.

Training diagnostics separate raw reward, applicability, gate passage, return, value, raw credit, normalized credit, and final mixture contribution. For search and answer actions, the signed contribution $n_k \bar{A}_k/(n_s + n_a)$ distinguishes a type's mean advantage from its frequency in the batch. Mean interaction turns, search-budget saturation, missing-answer frequency, and generated-token counts characterize behavior beyond EM. Candidate counts, fallback rates, missing values, scale-degenerate pools, anchor magnitudes, and clipping frequency help diagnose estimator failures. Observations of these statistics should be linked to the same run and training-step range before drawing a connection between credit and behavior.

F Interaction Interface and Qualitative Analysis

F.1 Action serialization

Each policy call receives the question and available interaction history, and emits one action after a think block. Search actions request external information; answer actions submit the final answer. Retrieved content is returned in an information block and becomes available to subsequent actions. The interface below summarizes the expected serialization, with braces denoting variable fields. It specifies the action structure rather than reproducing a full system or user prompt.

Search action

```
<think>{reasoning}</think>
<search>{query}</search>
```

Environment observation

```
<information>{retrieved documents}</information>
```

Answer action

```
<think>{reasoning}</think>
<answer>{final answer}</answer>
```

F.2 Reading a trajectory through its rewards

A trajectory-level analysis distinguishes action execution from reward eligibility. A search can return documents but receive no process credit if its format condition fails. A later search that only repeats earlier documents receives redundancy score -1 and no new evidence gain, even if the trajectory ultimately answers correctly. A correct answer with incomplete format can receive a positive raw outcome reward, but its outcome advantage and trajectory anchor are excluded at that failing turn; only format credit remains eligible. These cases follow from the evaluator and routing definitions; they are not reported as sampled model trajectories.

For an empirical case, the useful unit is a complete question–trajectory pair with the raw action, projected action, retrieved document identifiers or normalized texts, active masks, four reward scores, gate decisions, and final answer at each turn. This makes it possible to distinguish missing supervision, tool failure, repeated evidence, and a genuine credit imbalance.

G SUPPLEMENTARY TRAINING DIAGNOSTICS

G.1 ACTION-TYPE ISOLATION AND INTERACTION COLLAPSE

Figure 5 compares full TRACER with the removal of action-type isolation on Qwen2.5-1.5B under the same reward configuration, GAE setting, interaction budget, and seed. The mixed-pool variant assigns search actions persistently lower mean final advantages than answer actions. Its mean interaction turns drop toward one and remain there, whereas full TRACER sustains multi-turn interaction. Panel (c) shows that trajectories with no search calls become nearly universal, identifying the collapse as direct answering rather than one search followed by an answer.



(a) Advantage gap (b) Interaction turns (c) Zero-search fraction

Figure 5: Action-type isolation, training steps 1–654. (a) Search–answer advantage gap $D = \bar{A}_{\text{search}} - \bar{A}_{\text{answer}}$; negative values favor answers. (b) Mean interaction turns. (c) Zero-search trajectory fraction. Curves use trailing 50-step windows; faint lines in (b–c) show raw batches.

Statistics. In Figure 5(a), $\bar{A}_k = \sum_{j \in W} n_{k,j} \mu_{k,j} / \sum_{j \in W} n_{k,j}$ pools logged mean final advantages $\mu_{k,j}$ using action counts $n_{k,j}$ over up to 50 training steps; empty action groups are excluded. Panel (b) averages batch mean interaction turns over the same window. Panel (c) reports $\sum_{j \in W} N_{0,j} / \sum_{j \in W} N_j$, where $N_{0,j}$ counts trajectories with zero executed search calls and N_j counts all sampled trajectories at training step j. Both counts come from the logged tool-call histogram; the denominator includes every trajectory, regardless of answer correctness.

Connection to validation performance. Mixed comparison pools can turn differences between action-type-specific reward distributions into a preference for answering rather than gathering evidence. The early negative advantage gap and subsequent loss of retrieval are consistent with this mechanism and with the slow improvement and low validation plateau in Figure 4(e). The diagnostic concerns the behavior associated with the ablation; an action-conditioned mean alone does not distinguish action quality from a systematic estimator bias. Search actions also become rare after collapse, so their late-stage means summarize fewer observations.

H Additional Multi-Reward Baseline Comparisons

H.1 Reward-wise normalization with GDPO

GDPO normalizes each reward component within a question's trajectory group, combines the resulting credits, and applies batch-level advantage normalization (Liu et al., 2026). Its original study includes tool calling and reward-side conditioning, so a comparison should not assume that GDPO ignores constraints or is inherently unsuitable for agents. Our multi-turn adapter aggregates each reward pathway into a trajectory score using the same discount and reward-side format check as GRPO, but normalizes the components separately before fusion. The resulting trajectory advantage is broadcast to the generated actions. The model, reward coefficients, trajectories per question, interaction budget, and evaluation protocol are otherwise matched at Qwen2.5-1.5B.

What the comparison tests. This comparison asks whether preserving reward distinctions through normalization alone is sufficient when feedback has different temporal scopes and applies to different action types. It also distinguishes conditioning a reward before estimation from routing its estimated advantage: reward-side masking changes both scores and their comparison statistics, whereas TRACER selects normalized pathway credits at each action and suppresses the trajectory anchor when a condition fails. The two operations are not interchangeable, since a zeroed reward can still have nonzero relative credit after centering.

Table 9: Qwen2.5-1.5B test EM (%). Macro averages seven benchmarks; bold marks the best result.

Method	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	Macro
GRPO	40.39	57.84	44.09	34.10	31.23	10.63	24.80	34.73
GDPO	40.75	56.79	41.66	37.16	28.47	12.49	28.80	35.16
TRACER	**44.29**	**59.50**	**45.20**	**40.89**	**34.40**	**14.56**	**32.80**	**38.81**



Figure 6: Qwen2.5-1.5B validation EM on NQ/HotpotQA. GDPO improves over GRPO but plateaus below TRACER under the same multi-reward search setting.

Results and scope. GDPO improves over GRPO in Table 9, while TRACER retains the stronger overall test performance. Figure 6 shows the same ordering in validation convergence: separate normalization helps, but does not close the gap to typed, action-aware credit estimation. This result concerns the specified multi-turn search adapter, rather than establishing a general limitation of GDPO across agentic tasks.

H.2 TRAJECTORY-MEAN CALIBRATION OF PROXMO

ProxMO-C is a scalar-reward control designed to separate reward accumulation from finer credit assignment. It retains ProxMO's proximity-based estimator, comparison pools, and success-rate modulation, but averages the process and format rewards within each trajectory before combining them with the terminal outcome reward. The process signals are retrieval redundancy and evidence gain, with unchanged applicability rules and weights.

Let $v_{g,t}$ mark a valid, nonpadding action and let $\tilde{r}_{g,t}^{(k)}$ be the baseline reward after its reward-side format check for process pathways; format itself retains its original score. For each auxiliary pathway $k \in \{f, d, e\}$, define

$$\begin{aligned} N_g^{(k)} &= \sum_{t=0}^{T_g-1} v_{g,t} b_{g,t}^{(k)}, \qquad \bar{r}_g^{(k)} = \frac{\sum_{t=0}^{T_g-1} v_{g,t} b_{g,t}^{(k)} \tilde{r}_{g,t}^{(k)}}{\max(1, N_g^{(k)})}, \\ R_g^{\mathrm{C}} &= c_o r_{g,T_g-1}^{(o)} + \sum_{k \in \{f,d,e\}} c_k \bar{r}_g^{(k)}, \qquad r_{g,t}^{\mathrm{C}} = \mathbb{I}[t = T_g - 1] R_g^{\mathrm{C}}. \end{aligned} \tag{25}$$

An inapplicable pathway contributes zero, without redistributing its weight. The denominator counts applicable actions before the format check, so invalid formatting cannot remove a process-scored action from the average and inflate the remaining score. The coefficients remain $c_o = 1$, $c_f = 0.625$, $c_d = 0.0625$, and $c_e = 0.3125$. The scalar score is emitted once at termination and propagated with the same 0.95 discount as ProxMO; it is not repeatedly added at every action.

Effect and remaining limitations. For bounded auxiliary signals, averaging prevents their magnitude from growing simply because more turns are scored. This reduces the accumulation pressure that can favor a repeatedly rewarded action type and makes the coefficient budget more comparable to that of the outcome reward. It does not enforce equal gradient contributions or remove all action preferences: discounting, task success, and mixed comparison pools can still depend on behavior. Moreover, averaging discards when a useful or invalid action occurred, and scalar fusion permits compensation between requirements before advantages are estimated.

What the comparison establishes. The gains of ProxMO-C over ProxMO in Table 1 show that reward calibration accounts for part of the improvement in this setting. Its lower plateau than TRACER in Figure 3 indicates that correcting accumulation alone does not recover the benefits of typed credit estimation and condition-gated routing. This is a control of the complete calibration recipe, including terminal relocation; it does not isolate averaging from reward timing or establish the contribution of each TRACER component separately.