SELF-INDUCED OUTCOME POTENTIAL: TURN-LEVEL CREDIT ASSIGNMENT FOR AGENTS WITHOUT VERIFIERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon LLM agents succeed or fail through intermediate information-gathering turns, yet training signals are usually observed only at the final answer. Turn-level shaping methods reward turns that increase the likelihood of a gold answer, but require answer supervision or stable task-specific verifiers that are often unavailable in agentic settings. Label-free reinforcement learning (RL) methods extract self-signals from output distributions, but mostly at the answer or trajectory level and therefore cannot attribute value to intermediate turns. We propose *Self-Induced Outcome Potential* (SIOP), which treats semantic clusters of final answers as latent future outcome states for potential-based turn-level credit assignment. For each query, we sample multiple rollouts, cluster their final answers into semantic outcome modes, and construct a reliability-aware target distribution over these latent states. Each turn is then rewarded by how much it increases support for reliable outcome states, and these rewards are propagated through turn-level advantages rather than a single rollout-level advantage broadcast to all turns. We formalize the framework and establish its connection to supervised turn-level shaping in the gold-answer limit. We evaluate SIOP on seven search-augmented agentic reasoning benchmarks with Qwen3 and Phi-4-mini backbones, and on multi-turn tool-integrated reasoning. SIOP achieves the best average performance among verifier-free methods at two model scales while approaching a gold-supervised outcome baseline.

1 INTRODUCTION

Recent reasoning models such as OpenAI o1 (OpenAI, 2026) and DeepSeek-R1 (Guo et al., 2025) have made a broader shift visible: large language models (LLMs) are increasingly trained to solve problems through extended reasoning rather than immediate answer generation. Agentic settings extend this paradigm from internal reasoning traces to external interaction. Instead of answering from a fixed prompt, an agent reasons over a partial state, retrieves evidence, calls tools, observes returned information, and decides whether further interaction is needed.

Agentic retrieval-augmented generation (RAG) systems surveyed by Li *et al.* (Li et al., 2025) frame retrieval as adaptive interaction rather than static preprocessing. Kimi K2 (Kimi Team, 2026) uses large-scale agentic data synthesis and joint reinforcement learning through real and synthetic environments. ReSearch (Chen et al., 2025b) and Search-R1 (Jin et al., 2025) train search-augmented models to interleave reasoning and retrieval. ToolRL (Qian et al., 2025) and SimpleTIR (Xue et al., 2026) study tool-use RL and tool-integrated reasoning under multi-turn interaction. However, many of these training recipes still rely on outcome-level feedback: the policy receives rewards mainly from final-answer correctness, task success, or terminal score after the full trajectory is complete. In long trajectories, a single terminal reward cannot reliably identify whether a specific thought, tool call, or observation improved the agent's future prospects. For example, ReasonRAG (Zhang et al., 2025c) and InForage (Qian & Liu, 2025) report that outcome-supervised agentic RL suffers from low exploration efficiency, gradient conflict, and sparse rewards.

To address this problem, one line of work proposes turn-level reward shaping. For example, methods such as IGPO (Wang et al., 2026), ΔBelief-RL (Auzina et al., 2026), TIPS (Xie et al., 2026), and

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.

InfoPO (Kong et al., 2026) measure whether an interaction turn increases the model's support for a desired target, such as the gold answer or the correct next action. However, these methods inherit a strong assumption: a rewardable target must already be available. In many agentic tasks, there is no unique gold answer or stable task-specific verifier against which a turn-level potential can be computed.

Another line of work removes this requirement by extracting training signals directly from model outputs. EMPO (Zhang et al., 2025b) clusters semantically equivalent responses and optimizes semantic entropy without external supervision, while TTRL (Zuo et al., 2025) performs test-time reinforcement learning by selecting the majority answer from repeated samples as a self-generated pseudo-target. Related verifier-free RL methods, including RLPR (Yu et al., 2025b), NOVER (Liu et al., 2025c), and DRO (Xu et al., 2026b), similarly show that model-internal confidence or answer distributions can support learning beyond tasks with explicit verifiers. Yet these methods remain mostly outcome-level: they tell the model which final responses appear more reliable, but not which intermediate turns actually moved the trajectory toward those outcomes.

Building on these gaps, this paper studies how to lift unlabeled outcome-level reliability signals into turn-level credit assignment. Our key abstraction is to cluster sampled final answers into *semantic outcome modes*, where each mode is a query-specific group of answers that express the same underlying meaning, and then treat these modes as *latent future states*. Even without gold answers or stable task-specific verifiers, a batch of rollouts for the same query induces a structured distribution over these latent states. We then define a potential function over this self-induced latent state space and reward each turn by how much it increases the model's support for reliable future outcome states. The resulting *self-induced outcome potential* generalizes information-potential shaping from gold-answer supervision to verifier-free settings without task-specific gold verifiers.

Concretely, we sample multiple full rollouts for each query, cluster their final answers into semantic outcome modes, and assign soft reliability weights to these modes, which yields a distributional target rather than a hard pseudo-label. Because estimating the full mode distribution at every turn is expensive, we instantiate this principle in practice with a multi-reference cluster-level approximation that conditions each rollout on its realized semantic outcome mode while scoring a small set of reference answers rather than a single prototype. Terminal outcome mass is added only at the final answer turn and propagated backward through turn-level advantages, so different turns of the same rollout receive different credit. Our contributions are summarized as follows.

1. We formulate *verifier-free turn-level credit assignment* as potential-based learning over self-induced semantic outcome modes, where clustered final answers serve as latent future states rather than fixed gold targets or trajectory-level pseudo-labels.
2. We define a *self-induced outcome potential* (SIOP) over these latent states, calibrating empirical outcome modes by reliability signals and converting prefix-to-prefix changes in cluster support into turn-level process rewards.
3. We construct turn-specific advantages for long-horizon agents, assigning credit only to trainable tokens within each turn rather than broadcasting a trajectory-level score.
4. We show that the proposed reward coincides with gold-answer shaping on gold-reaching rollouts in the supervised limit and measures step-wise marginal support gain. We also present an empirical study on seven search-QA benchmarks showing that our method outperforms outcome-only verifier-free baselines on average and approaches a gold-supervised outcome RL baseline.

2 Related Work

Turn-level credit assignment. GRPO (Shao et al., 2024) provides group-relative outcome supervision, while IGPO (Wang et al., 2026), TIPS (Xie et al., 2026), and InfoReasoner (Hu et al., 2026) reward a turn by its change in support for a gold answer or gold semantic class. GiGPO (Feng et al., 2026) refines agent credit through episode-level and anchor-state group comparisons using an available task reward. SIOP addresses a complementary question: how to construct turn-level feedback when task-specific gold rewards are unavailable, rather than proposing a new clipping or optimization-stability mechanism.

Learning from unlabeled outcomes. Semantic entropy groups meaning-equivalent answers to characterize uncertainty (Farquhar et al., 2024; Kuhn et al., 2023). TTRL (Zuo et al., 2025) uses

majority answers as pseudo-targets, and EMPO (Zhang et al., 2025b) optimizes semantic consistency at the outcome level. RESTRAIN (Yu et al., 2026) further addresses spurious consensus. SIOP instead combines evidence-calibrated outcome weights with prefix-conditioned support changes, using sampled answer modes as targets for intermediate turns rather than only final-response selection. SIOP thus differs from each line on one axis: from turn-level shaping in the target (self-induced rather than gold), and from label-free RL in the granularity (turns rather than outcomes). Appendix A gives the extended discussion.

3 Problem Setup

We study an agentic setting where an agent receives a query q and interacts with an environment over multiple turns before producing a final answer. At turn t, the agent emits reasoning tokens, optionally issues a tool call, and receives an observation. We treat this as a single interaction segment

$$s_t = (\text{thought}_t, \text{tool}_t, \text{obs}_t), \tag{1}$$

so that a complete rollout can be written directly as

$$\tau = (q, s_1, \ldots, s_T, y). \tag{2}$$

For convenience, we write $\tau_{\le t} = (q, s_1, \ldots, s_t)$ for the rollout truncated after turn t. For each query, we sample K rollouts from the current policy π_θ:

$$\tau^{(k)} \sim \pi_\theta(\cdot \mid q), \qquad k = 1, \ldots, K, \tag{3}$$

where the k-th rollout can be written as

$$\tau^{(k)} = (q, s_1^{(k)}, \ldots, s_{T_k}^{(k)}, y^{(k)}). \tag{4}$$

Supervision mismatch. Supervised turn-level shaping methods such as ΔBelief-RL (Auzina et al., 2026), IGPO (Wang et al., 2026), and TIPS (Xie et al., 2026) assume access to a gold answer $y^\star$ or an equivalent target representation. In our setting, training does not observe a unique correct answer, a stable answer set, or a task-specific correctness verifier. The task-specific information available during learning consists of sampled rollouts from the current policy.

Verifier-free terminology. We use *verifier-free* in the task-specific sense: training requires no gold answers, task-specific correctness verifiers, or hand-designed rewards. A frozen off-the-shelf natural language inference (NLI) model (DeBERTa-v2-xlarge-MNLI (He et al., 2021)) serves as a task-agnostic semantic primitive for answer clustering and evidence scoring; it shapes the reward but never sees gold answers or benchmark labels (Appendix B).

Semantic outcome modes. Let $\mathcal{Y}_q = \{y^{(1)}, \ldots, y^{(K)}\}$ denote the set of answers sampled for query q. Following semantic entropy by Kuhn et al. (Kuhn et al., 2023), we cluster these answers by bidirectional semantic entailment under the query context. Writing $y \Rightarrow y'$ for "y semantically entails y' given q," as judged by an NLI model applied to the query-contextualized pair, we assign y and y' to the same cluster iff $y \Rightarrow y'$ and $y' \Rightarrow y$. Applying this criterion to $\mathcal{Y}_q$ yields a partition

$$\mathcal{C}(q) = \{c_1, \ldots, c_M\}, \tag{5}$$

where each cluster $c_m \subseteq \mathcal{Y}_q$ collects answers with the same meaning. We call each cluster a *semantic outcome mode*. We write $c(k) \in \mathcal{C}(q)$ for the cluster containing the final answer $y^{(k)}$ of rollout k.

Cluster statistics. For each cluster $c \in \mathcal{C}(q)$, let

$$S_c = \{k \in \{1, \ldots, K\} \mid c(k) = c\} \tag{6}$$

denote the set of sampled rollouts whose final answers fall into that semantic outcome mode. The corresponding empirical cluster mass is

$$m(c \mid q) = \frac{|S_c|}{K}. \tag{7}$$

We use $q_\theta(c \mid q)$ to denote the training target distribution over semantic outcome modes. Sec. 4 instantiates this distribution by calibrating the empirical cluster mass with reliability signals.

Goal. Our objective is to construct a turn-level reward $r_{t,\text{proc}}^{(k)}$ for each segment $s_t^{(k)}$ and convert these rewards into turn-specific learning signals for policy optimization. The next section defines the self-induced target distribution, outcome potential, and policy update used to compute these quantities.

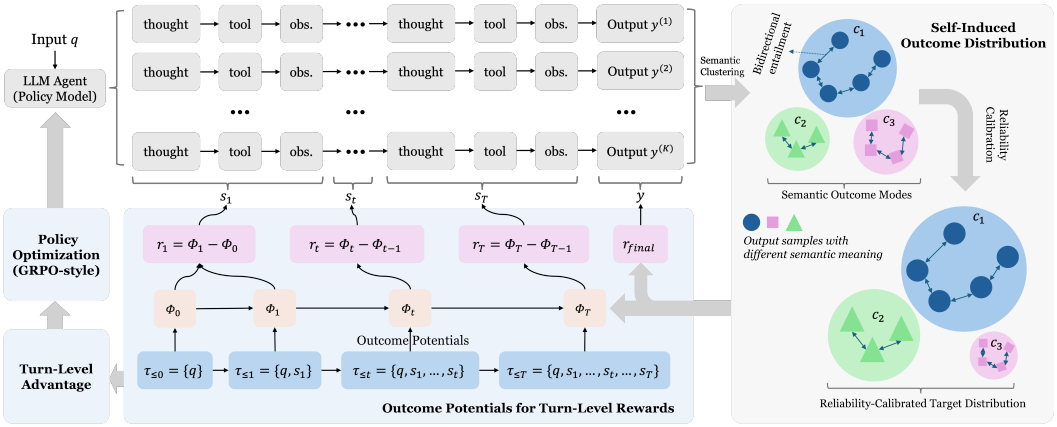


Figure 1: Overview of Self-Induced Outcome Potential (SIOP). For each query, the policy samples multiple agent rollouts, clusters their final answers into semantic outcome modes, and calibrates these modes into a reliability-weighted target distribution. SIOP then evaluates each prefix by its outcome potential, assigns turn rewards from potential differences, adds terminal outcome mass at the final answer turn, and updates the policy with turn-level advantages.

4 Methodology

SIOP starts from a credit-assignment question: without a gold answer or task-specific verifier, what future state should an intermediate turn be rewarded for moving toward? We answer this by using semantic outcome modes as the future-state space for potential-based shaping. Fig. 1 shows the resulting pipeline, which has four steps: build a reliability-calibrated target over semantic outcome modes (Sec. 4.1), score each turn by the change in outcome potential (Sec. 4.2), convert the turn rewards into turn-level advantages (Sec. 4.3), and update the policy with these advantages (Sec. 4.4). Algorithm 2 in Appendix C gives the full procedure, and Appendix B summarizes the notation.

4.1 Self-Induced Outcome Distribution

Semantic outcome modes. Multiple rollouts for the same query often produce answers that differ in surface form but express the same semantic hypothesis. Treating each string as a separate target makes the reward sensitive to wording, while selecting only the majority answer collapses uncertainty too early. We therefore group final answers into semantic outcome modes and treat each mode as a latent future state.

Let $S_c = \{k \mid c(k) = c\}$ denote the rollout subset assigned to cluster c. We begin with the empirical semantic outcome distribution

$$m(c \mid q) = \frac{|S_c|}{K}, \tag{8}$$

and let $R(c) \subseteq c$ denote a small reference set selected from cluster c, with nonnegative weights $\{\alpha_y^{(c)}\}_{y \in R(c)}$ satisfying

$$\sum_{y \in R(c)} \alpha_y^{(c)} = 1. \tag{9}$$

We take $R(c)$ to be the first N distinct non-empty surface forms in c, in rollout order, with uniform weights $\alpha_y^{(c)} = 1/|R(c)|$. These reference answers provide a tractable semantic proxy for the full cluster while avoiding commitment to a single canonical wording.

Reliability calibration. Empirical mode frequency is useful, but it is not a correctness guarantee. A wrong answer can become the majority when the base policy has a consistent bias or when retrieval returns misleading evidence. SIOP therefore does not directly optimize raw cluster frequency. It reweights the rollout-induced distribution using reliability evidence, so evidence can shift target mass away from a frequent but weakly supported mode toward better-supported alternatives.

We then calibrate the empirical distribution with a reliability correction

$$u_\theta(c, q) = \eta\, r(c, q), \tag{10}$$

where $\eta \geq 0$ controls the correction strength and $r(c, q)$ is a domain-appropriate reliability signal. The framework only requires that larger $r(c, q)$ indicate more reliable outcome modes.

For tool-using agents, we instantiate r as *evidence support*, namely the average degree to which tool observations along trajectories support the cluster's reference set:

$$r(c, q) = \frac{1}{|S_c|} \sum_{k \in S_c} \frac{1}{T_k} \sum_{t=1}^{T_k} \sum_{y \in R(c)} \alpha_y^{(c)} s_{\mathrm{NLI}}\big(\mathrm{obs}_t^{(k)},\, y\big), \tag{11}$$

where $s_{\mathrm{NLI}}(\mathrm{obs}, y)$ is the entailment confidence from an NLI model. In reasoning-only settings without tool observations, $r(c, q)$ can instead be instantiated with cluster-level self-consistency or mean log-confidence over the same reference set.

The resulting self-induced target distribution is

$$q_\theta(c \mid q) = \frac{m(c \mid q) \exp(u_\theta(c, q))}{\sum_{c' \in \mathcal{C}(q)} m(c' \mid q) \exp(u_\theta(c', q))}. \tag{12}$$

If $u_\theta(c, q) = 0$ for all clusters, this reduces to the empirical semantic outcome distribution induced by the sampled rollouts. Hence the self-induced target distribution can be viewed as the model's own semantic answer distribution, reweighted by reliability signals before it is used for turn-level shaping.

4.2 Semantic Outcome Potential for Turn-Level Rewards

Multi-reference cluster support. After forming reliable outcome modes, turn credit requires a prefix-conditioned support score for each mode. Because agentic answer spaces make exact mode probabilities impractical, we approximate mode support using a small reference set selected from the cluster.

The framework evaluates a turn against the distribution of reliable future outcome modes. For a truncated rollout $\tau_{\leq t}$, the true posterior mass on semantic outcome mode c is

$$p_\theta(c \mid \tau_{\leq t}) = \sum_{y \in \mathcal{Y}(q)} p_\theta(y \mid \tau_{\leq t})\, \mathbf{1}[y \in c]. \tag{13}$$

Because summing over the full answer space is intractable, we instantiate a tractable multi-reference surrogate for cluster support:

$$\tilde{p}_\theta(c \mid \tau_{\leq t}) = \sum_{y \in R(c)} \alpha_y^{(c)} p_\theta(y \mid \tau_{\leq t}), \tag{14}$$

which scores a small set of reference answers rather than the full cluster. When $R(c)$ contains a single answer, this recovers the earlier single-prototype approximation as a special case. With multiple references, however, the reward is less sensitive to any one wording choice and better reflects support for the semantic mode as a set.

Potential-based turn reward. Once reliable semantic modes and prefix-conditioned support are defined, SIOP assigns turn-level credit by measuring changes in a potential over the rollout's realized semantic outcome mode $c(k)$. A turn receives positive credit when it increases prefix support for this mode and negative credit when it decreases that support. The *outcome potential* at turn t is

$$\Phi_\theta(\tau_{\leq t}^{(k)}, c(k)) = q_\theta(c(k) \mid q)\, \log\big(\tilde{p}_\theta(c(k) \mid \tau_{\leq t}^{(k)})\big), \tag{15}$$

The potential combines a reliability-calibrated target mass with prefix-conditioned support. The target mass $q_\theta(c(k) \mid q)$ reflects the rollout frequency of the realized cluster after calibration by evidence support, while the log-support term measures how strongly the current prefix predicts that cluster. Thus potential differences are largest when a turn increases support for a cluster with high calibrated target mass. The step-level process reward is:

$$\begin{aligned} r_{t,\mathrm{proc}}^{(k)} &= \Phi_\theta(\tau_{\leq t}^{(k)}, c(k)) - \Phi_\theta(\tau_{\leq t-1}^{(k)}, c(k)) \\ &= q_\theta(c(k) \mid q)\Big[\log\big(\tilde{p}_\theta(c(k) \mid \tau_{\leq t}^{(k)})\big) - \log\big(\tilde{p}_\theta(c(k) \mid \tau_{\leq t-1}^{(k)})\big)\Big]. \end{aligned} \tag{16}$$

This reward is positive when turn t increases conditional support for the rollout's semantic outcome mode, and negative when it moves the trajectory away from that mode. All rollouts assigned to the same cluster share the same semantic reference set $R(c)$, so the reward operates at the cluster level rather than the individual answer level.

Properties. Two properties follow directly from this potential-difference form (Appendix D). First, the process rewards telescope: their sum over a rollout equals the total change in outcome potential from the query to the final state, so each turn is credited with its own marginal contribution (Proposition 1). Second, the reward factors into the calibrated target mass times the cluster-level support gain, so turns leading to low-mass modes receive little credit; when the target concentrates on the gold cluster with the gold answer as its only reference, the reward coincides with gold-answer support shaping on rollouts that reach the gold cluster, while other rollouts receive no process credit (Proposition 2). Appendix D.3 gives a Bayesian reading of the calibration step and proves when reliability evidence can reverse a spurious majority.

4.3 Turn-Level Advantage Construction

Process rewards identify which turns change semantic support, but global answer quality is still observed at the trajectory endpoint. We encode these two signals in an augmented immediate reward: process credit is assigned at every turn, while terminal outcome mass is added only at the final answer turn. For rollout k, this reward is

$$\bar{r}_t^{(k)} = \lambda r_{t,\text{proc}}^{(k)} + \mathbf{1}[t = T_k](1-\lambda) r_{\text{final}}^{(k)}, \tag{17}$$

where $r_{\text{final}}^{(k)} = q_\theta(c(k) \mid q)$ is the target-distribution weight of the rollout's semantic outcome mode. This design gives the final answer turn a direct outcome signal while allowing earlier turns to benefit only through return propagation.

For each query group (the K rollouts of one prompt), we pool all augmented rewards across its rollouts and turns and compute group-normalized immediate rewards using the group's own statistics:

$$\tilde{r}_t^{(k)} = \frac{\bar{r}_t^{(k)} - \mu_g}{\sigma_g}, \quad \text{where } \mu_g = \frac{1}{N_g} \sum_{k,t} \bar{r}_t^{(k)} \text{ and } \sigma_g = \sqrt{\frac{1}{N_g} \sum_{k,t} \left(\bar{r}_t^{(k)} - \mu_g\right)^2}, \tag{18}$$

with $N_g = \sum_{k=1}^{K} T_k$, the total number of (rollout, turn) pairs in that query group. Normalization is computed per query group (GRPO-style) rather than batch-globally, so different queries do not interfere with each other's turn scales. Pooling all turns within a group also means the z-score does not artificially rescale each turn index to the same variance: turns with systematically larger rewards retain their relative magnitudes within the group.

To propagate this terminal outcome signal back to earlier turns, we define the turn advantage as a discounted return over normalized immediate rewards:

$$A_t^{(k)} = \sum_{u=t}^{T_k} \gamma^{u-t} \tilde{r}_u^{(k)}. \tag{19}$$

This construction resolves the core mismatch with broadcasted rollout-level advantages: the final reward is not copied to all tokens, but only affects earlier turns through the discounted return of turn-local normalized rewards.

4.4 Turn-Conditioned Policy Optimization

To use turn-level advantages in a token-level clipped objective, we first identify the policy tokens generated by each interaction turn. Let $\mathcal{T}_t^{(k)}$ denote the set of trainable policy tokens generated within turn t of rollout k, excluding prompts, previous context replay, tool observations, and other non-policy text. We assign the same turn advantage $A_t^{(k)}$ to all trainable tokens in $\mathcal{T}_t^{(k)}$, but not across different turns.

With token-level importance ratio $\rho_{t,j}^{(k)}$ for token $j \in \mathcal{T}_t^{(k)}$, we minimize a GRPO-style clipped objective:

Table 1: Main comparative results, reported as Exact Match and token-level F1 for 4B and 8B backbones. Gold-supervised baselines are shown in gray and are not bolded. TTRL, EMPO, and SIOP use no task-specific verifier. **Bold** marks the best verifier-free result within each metric block.

Method	Single-Hop QA			Multi-Hop QA				Avg.
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	
Exact Match Accuracy (4B)								
GRPO (gold-sup.)	0.283	0.623	0.436	0.330	0.340	0.121	0.464	0.371
IGPO (gold-sup.)	0.262	0.598	0.449	0.359	0.340	0.113	0.464	0.369
TTRL	0.262	0.607	0.416	0.318	0.314	0.090	0.400	0.344
EMPO	0.270	**0.611**	0.412	0.330	0.301	0.090	**0.456**	0.353
SIOP	**0.281**	0.604	**0.416**	**0.344**	**0.367**	**0.100**	0.408	**0.360**
Token-level F1 (4B)								
GRPO (gold-sup.)	0.408	0.703	0.478	0.441	0.409	0.183	0.534	0.451
IGPO (gold-sup.)	0.396	0.682	0.486	0.451	0.381	0.184	0.550	0.447
TTRL	0.402	0.691	0.458	0.401	0.357	0.157	0.520	0.427
EMPO	**0.404**	**0.696**	0.460	0.414	0.347	0.159	0.510	0.427
SIOP	0.400	0.691	**0.465**	**0.443**	**0.411**	**0.179**	**0.528**	**0.445**
Exact Match Accuracy (8B)								
GRPO (gold-sup.)	0.270	0.643	0.418	0.391	0.363	0.141	0.488	0.388
IGPO (gold-sup.)	0.275	0.641	0.461	0.402	0.393	0.178	0.456	0.401
TTRL	0.266	0.623	0.416	0.385	0.357	**0.133**	0.440	0.374
EMPO	0.251	0.630	0.415	**0.392**	0.357	0.120	0.472	0.377
SIOP	**0.270**	**0.646**	**0.445**	0.348	**0.412**	0.107	**0.488**	**0.388**
Token-level F1 (8B)								
GRPO (gold-sup.)	0.422	0.728	0.474	0.484	0.416	0.224	0.599	0.478
IGPO (gold-sup.)	0.407	0.730	0.498	0.499	0.461	0.276	0.583	0.493
TTRL	0.401	0.722	0.476	0.431	0.416	0.192	0.539	0.454
EMPO	**0.406**	0.717	0.450	0.445	0.420	0.193	0.580	0.459
SIOP	0.403	**0.736**	**0.479**	**0.453**	**0.474**	**0.203**	**0.612**	**0.480**

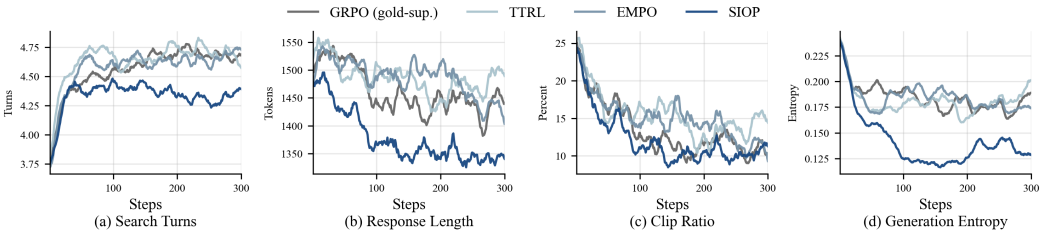


Figure 2: Training dynamics.

$$\mathcal{L}_{\text{SIOP}} = -\mathbb{E}\left[\sum_{k=1}^{K}\sum_{t=1}^{T_k}\sum_{j\in\mathcal{T}_t^{(k)}} \min\left(\rho_{t,j}^{(k)}A_t^{(k)}, \text{clip}(\rho_{t,j}^{(k)}, 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}})A_t^{(k)}\right)\right] + \beta\,\text{KL}(\pi_\theta \,\|\, \pi_{\theta_{\text{old}}}). \tag{20}$$

We refer to this objective as the *SIOP objective*. It preserves the main appeal of group-based RL, namely critic-free optimization with normalized on-policy signals, while ensuring that the optimization signal remains turn-specific. Within a turn, tokens share the same local credit, but advantages differ across turns. Full training pseudocode is deferred to Appendix C.

5 Experiments

5.1 Experimental Settings

Tasks and benchmarks. Our primary evaluation setting is search-augmented multi-turn question answering, which is the cleanest testbed for the proposed method because intermediate retrieval decisions are essential. Following this line of work, we train on a merged search-QA environment built from the Natural Questions (NQ) (Kwiatkowski et al., 2019), 2WikiMultiHopQA (Ho et al., 2020), and TriviaQA (Joshi et al., 2017) training splits and evaluate on seven benchmarks: NQ, TriviaQA (Joshi et al., 2017), HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), Bamboogle (Press et al., 2023), and PopQA (Mallen et al., 2023).

Table 2: Ablations on Qwen3-4B. The first four ablation rows remove one ingredient from *Full SIOP*: multi-reference cluster support, reliability calibration, both together, or the per-turn process reward. The last row keeps the SIOP reward and replaces the turn-level advantage with a rollout-level advantage broadcast to all turns. The agent loop, retriever, and optimizer are kept identical across rows, so each row isolates a single design choice. **Bold** marks the best entry in each column.

Method	Single-Hop QA			Multi-Hop QA				Avg.
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	
Accuracy (Exact Match)								
Full SIOP	**0.281**	0.604	0.416	0.344	**0.367**	**0.100**	0.408	**0.360**
w/o multi-ref	0.266	0.604	0.408	0.318	0.328	0.090	0.384	0.343
w/o calibration	0.252	0.609	0.416	0.332	0.336	0.082	0.400	0.347
w/o multi-ref, w/o calib	0.256	0.596	0.418	0.311	0.313	0.076	0.407	0.340
w/o process reward	0.273	**0.627**	0.426	0.330	0.307	0.094	0.368	0.346
w/o turn-level advantage	0.252	0.602	**0.438**	**0.348**	0.287	0.098	**0.432**	0.351
Token-level F1								
Full SIOP	0.400	0.691	0.465	**0.443**	**0.411**	**0.179**	**0.528**	**0.445**
w/o multi-ref	**0.407**	0.699	0.463	0.405	0.379	0.155	0.481	0.427
w/o calibration	0.396	0.687	0.473	0.418	0.378	0.147	0.521	0.431
w/o multi-ref, w/o calib	0.403	0.687	0.459	0.401	0.354	0.144	0.522	0.424
w/o process reward	0.400	**0.703**	0.464	0.429	0.357	0.172	0.481	0.429
w/o turn-level advantage	0.391	0.680	**0.489**	0.428	0.333	0.164	0.510	0.428

Backbone and environment. We use Qwen3-4B (Yang et al., 2025) and Qwen3-8B (Yang et al., 2025) as the policy backbone and run a search-augmented agent with a fixed interface across methods: at each turn the model produces reasoning text, optionally issues a search query, and receives retrieved passages before deciding the next turn or final answer. The prompting template, retrieval backend, passages per turn, and maximum interaction budget are held fixed across baselines so comparisons isolate the reward design. Full hyperparameters are listed in Appendix B.

Metrics. We report Exact Match (EM) and token-level F1 on the final extracted answer. In addition, we track behavior-level diagnostics in Fig. 2: average number of search turns, response length, response-length clip rate, and generation entropy.

Baselines. SIOP is compared against two verifier-free baselines: TTRL (Zuo et al., 2025), which rewards only the dominant semantic cluster as a hard majority pseudo-target, and EMPO (Zhang et al., 2025b), which rewards each rollout by the empirical frequency of its answer's cluster. We additionally include two gold-supervised baselines for context: GRPO (Shao et al., 2024), which uses exact-match correctness of the gold answer as an outcome reward, and IGPO (Wang et al., 2026), which rewards each turn according to how much it increases the model's probability of the gold answer.

5.2 Comparative Results

Main result. On the seven search-QA benchmarks (Tab. 1), SIOP achieves the best average performance among verifier-free methods at both scales, improving over the stronger of TTRL and EMPO by 0.7 EM and 1.8 F1 points at 4B and by 1.1 EM and 2.1 F1 points at 8B. The larger F1 gains are consistent with soft cluster targets that reward equivalent surface forms. **Multi-hop benchmarks.** As supporting evidence for turn-level credit, SIOP has the best verifier-free F1 on all four multi-hop benchmarks at both scales, with the largest margin on 2Wiki (+5.3 and +5.5 EM at 4B and 8B). **Gold-supervised baselines.** These contextualize the remaining gap: without gold labels, SIOP trails the turn-level IGPO by 0.9 EM and 0.2 F1 points at 4B and by 1.3 and 1.3 points at 8B, and matches the average EM of the outcome-level GRPO at 8B. Without answer labels, SIOP also raises the average EM of Phi-4-mini-instruct (Microsoft et al., 2025) from 0.210 to 0.367 and the MATH500 accuracy of Qwen3-8B in multi-turn tool-integrated reasoning from 0.453 to 0.801 (Appendix E.2), and its calibrated target q_θ ranks semantic modes consistently with gold answers never used in training (cluster-level AUC 0.667 at 4B and 0.733 at 8B; Appendix E.3).

5.3 Training Dynamics Analysis

Fig. 2 tracks four behavior diagnostics over 300 optimization steps. SIOP first increases tool use and then stabilizes below the verifier-free baselines (a), produces shorter responses with lower clip ratios

Table 3: Sensitivity analysis on Qwen3-4B (average EM and F1 over the seven benchmarks). † marks the default setting. The rows η=0 and N=1 coincide with *w/o calibration* and *w/o multi-ref* in Tab. 2. The NLI model is replaced in both answer clustering and evidence scoring.

	Calibration strength η				References N		NLI model		
	0	0.5	1†	2	1	3†	DeBERTa†	RoBERTa	BART
Avg. EM	0.347	0.349	0.360	0.352	0.343	0.360	0.360	0.358	0.363
Avg. F1	0.431	0.436	0.445	0.440	0.427	0.445	0.445	0.449	0.440

(b, c), and reaches a lower-entropy generation regime (d). Overall, SIOP produces fewer search turns and shorter responses without lowering end-task accuracy.

5.4 Ablation Studies

The ablations test the two sources of SIOP's gains over outcome-level methods: a more reliable target (calibration and multi-reference support) and credit for the intermediate turns that increase support for that target (process reward and turn-level advantage).

Component ablation. Tab. 2 removes multi-reference cluster support, reliability calibration, both together, or the per-turn process reward (λ=0, so intermediate turns receive zero immediate reward while the calibrated terminal reward still flows backward through the turn-level advantage) from full SIOP. Dropping calibration, multi-reference, or both lowers average EM by 1.3, 1.7, and 2.0 points and F1 by 1.4, 1.8, and 2.1 points, showing that both structural ingredients matter. The two ingredients affect different benchmarks: removing multi-reference support causes the larger drop on HotpotQA (-2.6 EM, -3.8 F1), 2Wiki EM (-3.9), and Bamboogle F1 (-4.7), whereas removing calibration causes the larger drop on NQ EM (-2.9) and MuSiQue (-1.8 EM, -3.2 F1). Removing the process reward costs 1.4 EM and 1.6 F1 points on average, mainly on multi-hop bridging tasks (2Wiki -6.0 EM, Bamboogle -4.0 EM), so process rewards help most when credit spans several informative turns.

Credit-assignment mechanism. The last row of Tab. 2 keeps the SIOP per-turn process reward fixed and replaces only the turn-level advantage with a GRPO-style broadcast: per-turn rewards are summed into one trajectory scalar and copied to every token (*w/o turn-level advantage*). With the reward held fixed, broadcasting the advantage drops average EM from 0.360 to 0.351 and F1 from 0.445 to 0.428, close to the outcome-only EMPO baseline in Tab. 1 (0.353 EM / 0.427 F1). The drop concentrates on multi-hop benchmarks (2Wiki, MuSiQue), indicating that the gains come from turn-level credit assignment rather than denser reward shaping alone.

Sensitivity. Tab. 3 varies the three choices that shape the SIOP reward. Performance changes with the calibration strength: the default η=1 is best, η=0.5 and η=2 lower average F1 by 0.9 and 0.5 points, and removing calibration (η=0) gives the lowest F1. Using N=3 references instead of one improves average EM by 1.7 and F1 by 1.8 points, so averaging over several surface forms of a mode gives a more reliable support estimate than a single prototype. Replacing the NLI model, DeBERTa-v2-xlarge-MNLI (He et al., 2021), with RoBERTa-large-MNLI (Liu et al., 2019) or BART-large-MNLI (Lewis et al., 2020) in both clustering and evidence scoring keeps average EM within 0.358 to 0.363 and F1 within 0.440 to 0.449. This is consistent with SIOP converting the discrete NLI partition into graded credit rather than a hard pseudo-label.

6 Conclusion

In this work, we have introduced verifier-free turn-level credit assignment for agentic reinforcement learning and instantiated it with Self-Induced Outcome Potential (SIOP). SIOP treats clustered final answers as latent future states and rewards each turn by the prefix-conditioned support gain it produces for the rollout's realized mode, weighted by a reliability-calibrated belief over modes. On seven question-answering benchmarks with Qwen3-4B and Qwen3-8B backbones, SIOP outperforms outcome-only verifier-free baselines and approaches the average performance of a gold-supervised outcome baseline without using gold labels during training. Ablations show that both the reliability-calibrated target and turn-specific credit contribute to these gains. Appendix G discusses limitations, including the dependence on semantic clustering and evidence estimation. Extending outcome states beyond text-answer clusters, for example to structured tool traces, is a natural next step.

REFERENCES

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. The Unreasonable Effectiveness of Entropy Minimization in LLM Reasoning. In *Advances in Neural Information Processing Systems*, 2025.

Ilze Amanda Auzina, Joschka Strüber, Sergio Hernández-Gutiérrez, Shashwat Goel, Ameya Prabhu, and Matthias Bethge. Intrinsic Credit Assignment for Long Horizon Interaction, 2026.

Minghan Chen, Guikun Chen, Wenguan Wang, and Yi Yang. SEED-GRPO: Semantic Entropy Enhanced GRPO for Uncertainty-Aware Policy Optimization, May 2025a. arXiv:2505.12346 [cs].

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. ReSearch: Learning to Reason with Search for LLMs via Reinforcement Learning. In *Advances in Neural Information Processing Systems*, 2025b.

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. ReSearch: Learning to Reason with Search for LLMs via Reinforcement Learning, September 2025c. arXiv:2503.19470 [cs].

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning convertsweak language models to strong language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-Group Policy Optimization for LLM Agent Training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rStar-Math: Small LLMs Can Master Math Reasoning with Self-Evolved Deep Thinking. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Yiran Guo, Lijie Xu, Jie Liu, Ye Dan, and Shuang Qiu. Segment Policy Optimization: Effective Segment-Level Credit Assignment in RL for Large Language Models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In *International Conference on Learning Representations*, 2021.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Nuria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580.

Senkang Hu, Yong Dai, Yuzhi Zhao, Yihang Tao, Yu Guo, Zhengru Fang, Sam Tak Wu Kwong, and Yuguang Fang. Optimizing Agentic Reasoning with Retrieval via Synthetic Semantic Information Gain Reward. *arXiv preprint arXiv:2602.00845*, 2026.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-Zero: Self-Evolving Reasoning LLM from Zero Data. In *The Fourteenth International Conference on Learning Representations*, 2026.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. Tree Search for LLM Agent Reinforcement Learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

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 *Second Conference on Language Modeling*, 2025.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147.

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 *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Kimi Team. Kimi K2: Open Agentic Intelligence, 2026.

Fanqi Kong, Jiayi Zhang, Mingyi Deng, Chenglin Wu, Yuyu Luo, and Bang Liu. InfoPO: Information-Driven Policy Optimization for User-Centric Agents. In *ICLR 2026 Workshop on Lifelong Agents: Learning, Aligning, Evolving*, 2026.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation. In *The Eleventh International Conference on Learning Representations*, 2023.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, 2020.

Yangning Li, Weizhi Zhang, Yuyao Yang, Wei-Chieh Huang, Yaozu Wu, Junyu Luo, Yuanchen Bei, Henry Peng Zou, Xiao Luo, Yusheng Zhao, Chunkit Chan, Yankai Chen, Zhongfen Deng, Yinghui Li, Hai-Tao Zheng, Dongyuan Li, Renhe Jiang, Ming Zhang, Yangqiu Song, and Philip S. Yu. Towards Agentic RAG with Deep Reasoning: A Survey of RAG-Reasoning Systems in LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 12120–12145. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.648.

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*, 2024.

Bo Liu, Leon Guertler, Simon Yu, Zichen Liu, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, Wee Sun Lee, and Natasha Jaques. SPIRAL: Self-Play on Zero-Sum Games Incentivizes Reasoning via Multi-Agent Multi-Turn Reinforcement Learning. *arXiv preprint arXiv:2506.24119*, 2025a.

Bo Liu, Chuanyang Jin, Seungone Kim, Weizhe Yuan, Wenting Zhao, Ilia Kulikov, Xian Li, Sainbayar Sukhbaatar, Jack Lanchantin, and Jason Weston. SPICE: Self-Play In Corpus Environments Improves Reasoning. *arXiv preprint arXiv:2510.24684*, 2025b.

Wei Liu, Siya Qi, Xinyu Wang, Chen Qian, Yali Du, and Yulan He. NOVER: Incentive Training for Language Models via Verifier-Free Reinforcement Learning. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 7439–7458, Suzhou, China, November 2025c. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.378.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019.

Xinji Mai, Haotian Xu, Zhong-Zhi Li, Xing W, Weinong Wang, Jian Hu, Yingying Zhang, and Wenqiang Zhang. Agent RL Scaling Law: Agent RL with Spontaneous Code Execution for Mathematical Problem Solving, 2025.

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 Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546.

Microsoft, Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, Dong Chen, Dongdong Chen, Junkun Chen, Weizhu Chen, Yen-Chun Chen, Yi ling Chen, Qi Dai, Xiyang Dai, Ruchao Fan, Mei Gao, Min Gao, Amit Garg, Abhishek Goswami, Junheng Hao, Amr Hendy, Yuxuan Hu, Xin Jin, Mahmoud Khademi, Dongwoo Kim, Young Jin Kim, Gina Lee, Jinyu Li, Yunsheng Li, Chen Liang, Xihui Lin, Zeqi Lin, Mengchen

Liu, Yang Liu, Gilsinia Lopez, Chong Luo, Piyush Madan, Vadim Mazalov, Arindam Mitra, Ali Mousavi, Anh Nguyen, Jing Pan, Daniel Perez-Becker, Jacob Platin, Thomas Portet, Kai Qiu, Bo Ren, Liliang Ren, Sambuddha Roy, Ning Shang, Yelong Shen, Saksham Singhal, Subhojit Som, Xia Song, Tetyana Sych, Praneetha Vaddamanu, Shuohang Wang, Yiming Wang, Zhenghao Wang, Haibin Wu, Haoran Xu, Weijian Xu, Yifan Yang, Ziyi Yang, Donghan Yu, Ishmam Zabir, Jianwen Zhang, Li Lyna Zhang, Yunan Zhang, and Xiren Zhou. Phi-4-Mini Technical Report: Compact yet Powerful Multimodal Language Models via Mixture-of-LoRAs, 2025.

OpenAI. OpenAI o1 System Card, 2026.

Chunyi Peng, Zhipeng Xu, Zhenghao Liu, Yishan Li, Yukun Yan, Shuo Wang, Zhiyuan Liu, Yu Gu, Minghe Yu, Ge Yu, and Maosong Sun. Learning to Route Queries Across Knowledge Bases for Step-wise Retrieval-Augmented Reasoning, May 2025. arXiv:2505.22095 [cs].

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

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. ToolRL: Reward is All Tool Learning Needs. In *Advances in Neural Information Processing Systems*, 2025.

Hongjin Qian and Zheng Liu. Scent of Knowledge: Optimizing Search-Enhanced Reasoning with Information Foraging. In *Advances in Neural Information Processing Systems*, 2025.

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, 2024.

Hieu Tran, Zonghai Yao, and Hong Yu. Exploiting Tree Structure for Credit Assignment in RL Training of LLMs, 2025.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. ♪ MuSiQue: Multihop 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.

Quan Wei, Siliang Zeng, Chenliang Li, 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 Turn-Level Reward Design, 2025.

Yutao Xie, Nathaniel Thomas, Nicklas Hansen, Yang Fu, Li Erran Li, and Xiaolong Wang. TIPS: Turn-level Information-Potential Reward Shaping for Search-Augmented LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026.

Donglai Xu, Hongzheng Yang, Yuzhi Zhao, Pingping Zhang, Jinpeng Chen, Wenao Ma, Zhijian Hou, Mengyang Wu, Xiaolei Li, Senkang Hu, Ziyi Guan, Jason Chun Lok Li, and Lai Man Po. From Exploration to Exploitation: A Two-Stage Entropy RLVR Approach for Noise-Tolerant MLLM Training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026a.

Yifei Xu, Tusher Chakraborty, Srinagesh Sharma, Leonardo Nunes, Swati Sharma, Kate Drakos Demopulos, Emre Kıcıman, Songwu Lu, and Ranveer Chandra. Direct Reasoning Optimization: Constrained RL with Token-Level Dense Reward and Rubric-Gated Constraints for Open-ended Tasks, 2026b.

Zhenghai Xue, Longtao Zheng, Qian Liu, Yingru Li, Xiaosen Zheng, Zejun MA, and Bo An. SimpleTIR: End-to-End Reinforcement Learning for Multi-Turn Tool-Integrated Reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 Technical Report, 2025.

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 Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025a.

Tianyu Yu, Bo Ji, Shouli Wang, Shu Yao, Zefan Wang, Ganqu Cui, Lifan Yuan, Ning Ding, Yuan Yao, Zhiyuan Liu, Maosong Sun, and Tat-Seng Chua. RLPR: Extrapolating RLVR to General Domains without Verifiers, 2025b.

Zhaoning Yu, Zhaolun Su, Leitian Tao, Haozhu Wang, Aashu Singh, Hanchao Yu, Jianyu Wang, Hongyang Gao, Weizhe Yuan, Jason E Weston, et al. RESTRAIN: From Spurious Votes to Signals—Self-Training RL with Self-Penalization. In *The Fourteenth International Conference on Learning Representations*, 2026.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Hanning Zhang, Pengcheng Wang, Shizhe Diao, Yong Lin, Rui Pan, Hanze Dong, Dylan Zhang, Pavlo Molchanov, and Tong Zhang. Entropy-Regularized Process Reward Model, 2025a.

Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. Right Question is Already Half the Answer: Fully Unsupervised LLM Reasoning Incentivization. In *Advances in Neural Information Processing Systems*, 2025b.

Wenlin Zhang, Xiangyang Li, Kuicai Dong, Yichao Wang, Pengyue Jia, Xiaopeng Li, Yingyi Zhang, Derong Xu, Zhaocheng Du, Huifeng Guo, Ruiming Tang, and Xiangyu Zhao. Process vs. Outcome Reward: Which is Better for Agentic RAG Reinforcement Learning. In *Advances in Neural Information Processing Systems*, 2025c.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute Zero: Reinforced Self-play Reasoning with Zero Data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026a.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to Reason without External Rewards. In *International Conference on Learning Representations*, 2026b.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Xiangxin Zhou, Zichen Liu, Anya Sims, Haonan Wang, Tianyu Pang, Chongxuan Li, Liang Wang, Min Lin, and Chao Du. Reinforcing General Reasoning without Verifiers. In *International Conference on Learning Representations*, 2026.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. TTRL: Test-Time Reinforcement Learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Appendix

CONTENT

A. Related Work 14
A.1 Outcome RL and Supervised Step-Level Credit Assignment 14
A.2 Label-Free Reinforcement Learning from Output Distributions 15
A.3 Open-Ended Search and Tool-Using Agents 15
B. Implementation Details 16
C. Training Algorithm 17
D. Theoretical Characterizations and Proofs 19
D.1 Telescoping of the Outcome Potential 19
D.2 Factorization and Supervised Limit 20
D.3 Bayesian View and Reliability Calibration 21
E. Additional Results 22
E.1 Compute Overhead 22
E.2 Additional Backbones and Tasks 22
E.3 Semantic Modes and Target Quality 23
E.4 Case Study 24
F. Reward Hacking and Confident-but-Wrong Modes 24
G. Limitations and Future Work 25

A RELATED WORK

A.1 OUTCOME RL AND SUPERVISED STEP-LEVEL CREDIT ASSIGNMENT

Modern reasoning-model training typically starts from outcome-level RL with verifiable rewards. DeepSeekMath (Shao et al., 2024) introduced GRPO as a critic-free optimizer that compares sampled responses within each prompt group, and DeepSeek-R1 (Guo et al., 2025) showed that outcome rewards can elicit long chain-of-thought behavior at scale. DAPO (Yu et al., 2025a) and GSPO (Zheng et al., 2025) further improve this group-based paradigm by addressing sampling, clipping, loss normalization, and importance-weighting stability. This line clarifies how to optimize a policy when an outcome reward is already available. It does not by itself specify what the reward target should be without a task-specific verifier, or how that target should be assigned across interaction turns.

The complementary literature studies whether intermediate reasoning steps should receive their own supervision. Lightman et al. (Lightman et al., 2024) showed that process supervision can outperform outcome supervision for mathematical reasoning. Zhang et al. (Zhang et al., 2025a) generalized process reward models through entropy-regularized MDPs, and rStar-Math (Guan et al., 2025) uses MCTS-based self-evolved process preference models. Other methods refine the temporal granularity of RL itself: VinePPO (Kazemnejad et al., 2025) uses Monte Carlo rollouts to replace learned value functions for unbiased credit assignment, SPO (Guo et al., 2026) operates at a segment level between tokens and trajectories, TEMPO (Tran et al., 2025) derives nonparametric branch-gated corrections from prefix trees, and Zeng et al. (Wei et al., 2025) estimate turn-level advantages for multi-turn tool-use agents modeled as MDPs. These approaches show that finer credit is useful, but they still assume access to supervised step labels, verifiable outcomes, or an externally supplied task reward.

For multi-turn agents, information-potential methods make the dependence on a target especially explicit. IGPO (Wang et al., 2026) rewards a turn by the marginal increase in the policy's probability of the gold answer. ΔBelief-RL (Auzina et al., 2026) uses belief updates for long-horizon credit assignment. TIPS (Xie et al., 2026) defines an information-potential reward for search-augmented LLMs, InfoPO (Kong et al., 2026) adapts the idea to user-centric settings, and InfoReasoner (Hu et al., 2026) optimizes retrieval-augmented reasoning with a synthetic semantic information-gain reward over retrieval trajectories. These methods establish that turn-level shaping is important for long-horizon optimization, but their potentials are still anchored to a gold answer, supervised belief state, externally specified utility, or retrieval-specific information objective. SIOP changes the role

of the target: it constructs a query-specific target space from sampled final answers, treats semantic outcome modes as latent future states, and computes turn rewards as potential changes over this self-induced state space. The supervised gold-answer case is therefore a limiting case rather than the starting assumption (Proposition 2).

A.2 Label-Free Reinforcement Learning from Output Distributions

Label-free reasoning optimization tries to replace benchmark labels or task-specific verifiers with signals induced by the model's own samples. One family relies on agreement structure across multiple outputs. Semantic entropy by Kuhn et al. (Kuhn et al., 2023) and Farquhar et al. (Farquhar et al., 2024) estimates uncertainty over meaning-equivalent answer clusters, while TTRL (Zuo et al., 2025) uses majority behavior from repeated sampling as a self-generated label. EMPO (Zhang et al., 2025b) directly optimizes semantic clusters by minimizing semantic entropy, SEED-GRPO (Chen et al., 2025a) uses semantic entropy to modulate policy updates on high-uncertainty prompts, and Agarwal et al. (Agarwal et al., 2025) show that entropy minimization alone can be surprisingly effective. Xu et al. (Xu et al., 2026a) further study entropy scheduling from exploration to exploitation under noisy supervision. These methods establish that output distributions contain usable training signal, but they also expose a failure mode: high agreement is not always correctness. RESTRAIN (Yu et al., 2026) addresses this issue by penalizing spurious consensus rather than trusting majority votes unconditionally.

A second family uses internal model feedback, reference likelihood, or self-evaluation as the reward source. RLPR (Yu et al., 2025b) uses token probabilities assigned to reference answers, NOVER (Liu et al., 2025c) studies verifier-free incentive training from standard supervised data, DRO (Xu et al., 2026b) uses reasoning-reflection signals, VeriFree (Zhou et al., 2026) optimizes reference-answer likelihood without an external verifier, and Intuitor (Zhao et al., 2026b) replaces external rewards with self-certainty. Related self-improvement methods obtain training signal through self-rewarding or self-play: Self-Rewarding Language Models (Yuan et al., 2024) use LLM-as-a-Judge rewards, SPIN (Chen et al., 2024) frames alignment as self-play, SPIRAL (Liu et al., 2025a) and SPICE (Liu et al., 2025b) extend self-play to multi-turn settings, and Absolute Zero (Zhao et al., 2026a) and R-Zero (Huang et al., 2026) study self-evolution from zero external data.

The common limitation for our setting is not the absence of useful unlabeled signals, but their granularity. TTRL (Zuo et al., 2025), EMPO (Zhang et al., 2025b), SEED-GRPO (Chen et al., 2025a), RLPR (Yu et al., 2025b), NOVER (Liu et al., 2025c), VeriFree (Zhou et al., 2026), and Intuitor (Zhao et al., 2026b) primarily attach rewards to prompts, final answers, or complete trajectories. They help decide which sampled outcomes or prompts are more reliable, but they do not identify which intermediate interaction turns moved the trajectory toward those outcomes. SIOP uses the same kind of unlabeled distributional structure for a different object: semantic outcome modes define a query-specific latent future-state space, and process rewards are computed as potential changes over that space. This turns outcome-level reliability into turn-level credit assignment rather than only outcome selection.

A.3 Open-Ended Search and Tool-Using Agents

Search and tool-use agents make credit assignment harder than single-turn generation because the model must decide not only what to answer, but also when to retrieve, which evidence to trust, and when to stop interacting with the environment. The agentic RAG survey by Li et al. (Li et al., 2025) summarizes this shift from static retrieval to iterative information seeking. ReSearch (Chen et al., 2025b) and Scent of Knowledge (Qian & Liu, 2025) further show that search behavior itself can be optimized rather than treated as a fixed preprocessing step. Open-domain QA benchmarks such as PopQA (Mallen et al., 2023), HotpotQA (Yang et al., 2018), and MuSiQue (Trivedi et al., 2022) expose the same issue empirically: final-answer quality depends on intermediate evidence acquisition, especially for long-tail and multi-hop questions.

Reward design has therefore become central in tool-use and search-augmented RL. ToolRL (Qian et al., 2025) studies how reward granularity, scale, and temporal dynamics affect tool-use learning. Process vs. Outcome Reward (Zhang et al., 2025c) shows that finer-grained feedback can improve long-horizon agentic RAG, while also emphasizing the cost of obtaining gold intermediate supervision.

Table 4: Core notation used in the method section.

Symbol	Meaning
q	Input query.
K	Number of sampled rollouts for the same query.
$\tau^{(k)}, \tau^{(k)}_{\le t}$	Rollout k and its prefix after turn t.
T_k	Number of assistant turns in rollout k.
$y^{(k)}$	Final answer emitted by rollout k.
$c, c(k)$	Semantic outcome mode and the mode assigned to rollout k.
$\mathcal{C}(q)$	Semantic outcome modes induced by the sampled rollouts for query q.
S_c	Set of rollouts assigned to semantic mode c.
$R(c), \alpha^{(c)}_y$	Reference answers for mode c and their normalized weights.
$m(c \mid q)$	Empirical mass of mode c, computed from rollout counts.
$\mathrm{obs}^{(k)}_t$	Tool observation returned at turn t of rollout k.
$s_{\mathrm{NLI}}(\mathrm{obs}, y)$	Entailment confidence that observation obs supports answer y.
$r(c, q)$	Reliability score for mode c, instantiated as evidence support in search-agent experiments.
η	Strength of the reliability calibration applied to empirical mode mass.
$q_\theta(c \mid q)$	Reliability-calibrated target distribution over semantic outcome modes.
$\tilde{p}_\theta(c \mid \tau_{\le t})$	Multi-reference approximation to prefix-conditioned support for mode c.
$\Phi_\theta(\tau^{(k)}_{\le t}, c(k))$	Outcome potential for prefix $\tau^{(k)}_{\le t}$ and rollout k's mode.
$r^{(k)}_{t,\mathrm{proc}}$	Process reward from the change in outcome potential at turn t.
λ	Mixing coefficient between process reward and terminal outcome mass.
$\bar{r}^{(k)}_t$	Immediate turn reward after adding terminal outcome mass at the final answer turn.
γ	Discount factor used to propagate later rewards backward.
$\tilde{r}^{(k)}_t, A^{(k)}_t$	Group-normalized immediate reward and the discounted turn advantage used for policy optimization.
$\mathcal{T}^{(k)}_t$	Trainable policy tokens generated within turn t of rollout k.

These results support the need for turn-level feedback, but they still leave open how to obtain such feedback when neither intermediate labels nor task-specific answer verifiers are available.

Recent agentic RL methods adapt group-based optimization to multi-step environments. Search-R1 (Jin et al., 2025) and ReSearch (Chen et al., 2025c) train models to interleave reasoning with search, while GiGPO (Feng et al., 2026), Step-GRPO (Peng et al., 2025), and Tree-GRPO (Ji et al., 2026) construct group comparisons, step-level advantages, or tree-structured credit over agent trajectories. These methods improve optimization for search agents, but their learning signal still typically comes from a task reward, answer verifier, or environment-defined success criterion. SIOP addresses a complementary supervision gap. Instead of assuming that the environment can verify the final answer or score each tool call, it constructs a latent outcome space from unlabeled sampled answers and uses potential changes over that space to score intermediate turns. This lets the same output distribution that supports label-free outcome selection also guide which retrieval decisions and tool calls were informative.

B Implementation Details

Method notation. Table 4 lists the recurring symbols used in the method section. Local quantities used only in one equation are defined in place.

All training hyperparameters are summarized in Table 5, and the system prompt template is shown in Figure 3. The remainder of this subsection covers the three components of the pipeline that are not a single numeric value: training, retrieval, and semantic clustering.

Training. The policy backbones are Qwen3-4B and Qwen3-8B. The actor is trained with the verl framework (v0.8.0.dev) using FSDP and a hybrid-engine weight sync to a vLLM 0.11 rollout backend on 8× NVIDIA A800-80GB, with PyTorch 2.8.0+cu128 and flash-attention 2.8.1. Optimization uses AdamW with a constant learning rate of 10^{-6}, KL regularization to a frozen reference model (`low_var_kl` estimator, coefficient 0.005), discount factor γ=1, train batch size 128, and a PPO

mini-batch size of 32 with a micro-batch size of 4 per GPU. The SIOP hyperparameters are N=3 references per cluster, η=1 reliability calibration strength, and λ=0.5 process/terminal mixing. Each run is trained for 300 optimizer steps with validation every 50 steps. The compute profile in Table 6 is measured on Qwen3-4B, where a single SIOP run takes 11.75 wall-clock hours on 8×A800-80GB, or about 94 GPU-hours.

Retrieval. The retriever is a local Wikipedia index built from the `wiki-18` dump with `intfloat/e5-base-v2` sentence embeddings and a FAISS flat index over the full corpus. It runs as a standalone HTTP service with FAISS GPU acceleration, independent of the training processes, and each tool call returns the top-3 passages. Tool calls use the Hermes `<tool_call>`/`<tool_response>` format, with `max_response_length`=2048 and the tool response length capped at 2048 tokens per call.

Prompt. All methods share the same prompt template, as shown in Figure 3. A minimal system message sets the assistant role, and a user message specifies the interaction protocol: reasoning is wrapped in `<think>`/`</think>` tags after every new observation, search queries are issued with `<tool_call>`/`</tool_call>`, retrieval results return between `<tool_response>`/`</tool_response>`, and the final answer is emitted in `<answer>`/`</answer>`. The template allows up to five assistant turns and is identical across SIOP, TTRL, EMPO, and GRPO so that differences in end-task performance are not attributable to prompt engineering.

Semantic clustering. The framework is agnostic to the clustering backend: in structured settings, answers can be normalized by rules or regular expressions before clustering, and in open-ended settings a small judge model or a pairwise entailment model suffices, consistent with semantic-entropy style approaches by Kuhn et al. (Kuhn et al., 2023) and Farquhar et al. (Farquhar et al., 2024). In this paper we instantiate clustering with `microsoft/deberta-v2-xlarge-mnli` as a bidirectional-entailment judge. For each query's K rollouts we contextualize every answer $y^{(k)}$ with the query as "q $y^{(k)}$" before NLI scoring, so equivalence judgments depend on the question rather than surface form alone. Two contextualized answers a and b are merged when $a \Rightarrow b$ and $b \Rightarrow a$ under a strict bidirectional test: the NLI head must output the entailment class in *both* directions. We then run a one-pass greedy clustering over the K answers (Algorithm 1): each unassigned answer opens a new cluster, is batch-compared against all still-unassigned later answers, and the ones that bidirectionally entail it are absorbed. The same NLI model also produces the reliability signal $r(c, q)$ from retrieved context. It is loaded as a CPU-side singleton so it never contends with the vLLM rollout backend for GPU memory, and clustering is invoked through the same scoring service that computes $\tilde{p}_\theta$.

Reference sets. For each cluster, we traverse its members in rollout order, remove empty answers and exact-string duplicates, and keep the first N distinct surface forms as $R(c)$ (N=3 in all experiments). The first answer entering the cluster is its canonical reference and appears first in $R(c)$, but all references receive the same weight $\alpha_y^{(c)} = 1/|R(c)|$. With K=4 rollouts, a cluster has more than N distinct forms only when all four answers fall into it with different wordings. Each pairwise equivalence test is symmetric, so cluster membership does not depend on rollout order when the NLI judgments are transitive, and uniform averaging over several references reduces the influence of any single reference, including the canonical one.

Verifier status of the NLI model. Because its predictions shape the reward, the NLI model can be viewed as a weak semantic verifier under a broad definition that counts any learned scorer in reward construction. Under the task-verification definition used in this paper, it is not a verifier: it receives no gold answer, benchmark label, task-specific rule, or environment success signal, and cannot decide whether an answer is correct. Table 3 shows that results are stable across three NLI backbones.

C Training Algorithm

Algorithm 2 summarizes the full training pipeline. Each iteration processes one batch of queries in four phases. The *rollout and clustering* phase samples K full rollouts per query, clusters their final answers into semantic outcome modes via the NLI judge, and computes the reliability-calibrated target distribution $q_\theta(c \mid q)$ over those modes. The *scoring* phase runs a single forward pass on a frozen copy of the policy to obtain $\log p_\theta(y \mid \tau_{\leq t})$ for every $(y, \tau_{\leq t})$ pair, aggregates them into the

Algorithm 1 Greedy bidirectional-entailment clustering

Require: answers $\{y^{(k)}\}_{k=1}^{K}$, query q, NLI judge f_{NLI}
$\hat{y}^{(k)} \leftarrow$ "q $y^{(k)}$" for $k = 1, \ldots, K$ ▷ contextualize with query
$\mathrm{cid}[k] \leftarrow \perp$ for all k
$c \leftarrow 0$
for $k = 1, \ldots, K$ **do**
  if $\mathrm{cid}[k] \neq \perp$ **then**
    continue
  end if
  $\mathrm{cid}[k] \leftarrow c$
  Mark $y^{(k)}$ as the canonical reference for cluster c
  $\mathcal{U} \leftarrow \{j : j > k,\ \mathrm{cid}[j] = \perp\}$
  $e \leftarrow f_{\mathrm{NLI}}(\hat{y}^{(k)}, \hat{y}^{(j)})$ and $e' \leftarrow f_{\mathrm{NLI}}(\hat{y}^{(j)}, \hat{y}^{(k)})$ for all $j \in \mathcal{U}$ ▷ batched
  for each $j \in \mathcal{U}$ with $e_j =$ entail and $e'_j =$ entail **do**
    $\mathrm{cid}[j] \leftarrow c$
  end for
  $c \leftarrow c + 1$
end for
return cid, cluster canonical references

Table 5: Key hyperparameters used for SIOP training. Values apply to all ablations and baselines unless otherwise noted.

Hyperparameter	**Value**
Policy backbone	Qwen3-4B/8B
Training Batch Size	128
Validation Batch Size	128
Max Prompt Length	1024
Max Response Length	2048
PPO Mini-Batch Size	32
PPO Micro-Batch Size per GPU	4
Actor Learning Rate	1×10^{-6}
KL Loss Coefficient	0.005
KL Loss Estimator	low_var_kl
Discount Factor γ	1.0
Training Steps	300
Save / Validation Frequency	every 50 steps
Rollouts per Query K	4
Rollout Temperature	1.0
Max Assistant Turns (Training)	5
Max Tool Response Length	2048
Retrieval Top-k	3
Tensor Parallel Size (rollout)	4
Rollout GPU Memory Utilization	0.4
Scorer GPU Memory Utilization	0.15
Scorer Max Model Length	8192
SIOP Process-Terminal Mix λ	0.5
SIOP References per Cluster N	3
SIOP Reliability Strength η	1.0
NLI Strict Entailment	true
NLI Device	CPU
GPUs per Node	8

multi-reference cluster support $\tilde{p}_\theta(c \mid \tau_{\leq t})$, and turns consecutive differences into per-turn process rewards. The *advantage construction* phase adds terminal outcome mass at the final answer turn, pools the augmented rewards across the query group, computes the group-relative z-score, and propagates

Task Prompt Template

System: You are a helpful and harmless assistant.

User: Answer the given question. You must conduct reasoning inside `<think>` and `</think>` first every time you get new information. After reasoning, if you find you don't know the exact answer, you can just call a search engine by `<tool_call>` query `</tool_call>` (for example, suppose you don't know where the capital of china is, you can just output `<tool_call>capital china</tool_call>` to get a search) and it will return the top searched results between `<tool_response>` and `</tool_response>`. You can search as many times as you want. If you find no further external knowledge needed, you can directly provide the answer inside `<answer>` and `</answer>`, without detailed illustrations. For example, `<answer>` Beijing `</answer>`. Question: {question}.

Figure 3: Task prompt template used for training and inference. All rollouts see an identical system message and a user message that combines the retrieval-augmented reasoning instructions with the query.

it backward through the discounted turn-level advantage. The *policy update* phase broadcasts the turn-level advantage onto the trainable tokens within each turn and applies the GRPO-style clipped objective. The scoring pass is served by a GPU-side inference service sharing the rollout backend's memory budget, while the NLI calls run through a CPU-side singleton. Thus the main additional GPU cost over outcome-only RL is the forward pass over pseudo-response pairs at each training step.

D Theoretical Characterizations and Proofs

D.1 Telescoping of the Outcome Potential

Proposition 1 (Telescoping of the outcome potential)**.** *For any rollout $\tau^{(k)}$, let $\tau^{(k)}_{\le 0}$ denote the initial query state before agent-generated turns. The cumulative process reward satisfies*

$$\begin{aligned}\sum_{t=1}^{T_k} r^{(k)}_{t,\mathrm{proc}} &= \Phi_\theta(\tau^{(k)}_{\le T_k}, c(k)) - \Phi_\theta(\tau^{(k)}_{\le 0}, c(k)) \\ &= q_\theta(c(k) \mid q)\Big[\log\big(\tilde{p}_\theta(c(k) \mid \tau^{(k)}_{\le T_k})\big) - \log\big(\tilde{p}_\theta(c(k) \mid q)\big)\Big].\end{aligned} \tag{21}$$

Hence the process reward is potential-based: cumulative credit depends only on the change in outcome potential between the initial query state and the final trajectory state.

Proof. By definition of the outcome potential and the process reward,

$$\begin{aligned}r^{(k)}_{t,\mathrm{proc}} &= \Phi_\theta(\tau^{(k)}_{\le t}, c(k)) - \Phi_\theta(\tau^{(k)}_{\le t-1}, c(k)) \\ &= q_\theta(c(k) \mid q)\Big[\log\big(\tilde{p}_\theta(c(k) \mid \tau^{(k)}_{\le t})\big) - \log\big(\tilde{p}_\theta(c(k) \mid \tau^{(k)}_{\le t-1})\big)\Big].\end{aligned} \tag{22}$$

Summing over turns yields a telescoping series:

$$\sum_{t=1}^{T_k} r^{(k)}_{t,\mathrm{proc}} = \sum_{t=1}^{T_k}\Big[\Phi_\theta(\tau^{(k)}_{\le t}, c(k)) - \Phi_\theta(\tau^{(k)}_{\le t-1}, c(k))\Big], \tag{23}$$

and all interior terms cancel. The remaining boundary terms are

$$\begin{aligned}&\Phi_\theta(\tau^{(k)}_{\le T_k}, c(k)) - \Phi_\theta(\tau^{(k)}_{\le 0}, c(k)) \\ &\qquad = q_\theta(c(k) \mid q)\Big[\log\big(\tilde{p}_\theta(c(k) \mid \tau^{(k)}_{\le T_k})\big) - \log\big(\tilde{p}_\theta(c(k) \mid q)\big)\Big].\end{aligned} \tag{24}$$

Thus the cumulative reward depends on the improvement in final conditional support, while each turn's local credit is determined only by its own marginal contribution to that improvement.

Algorithm 2 Training with SIOP

Require: Query batch $\mathcal{Q}$, policy π_θ, frozen scoring copy, rollout count K, discount γ, mixture weight λ

for each query $q \in \mathcal{Q}$ **do**
  // Phase 1: Rollout and semantic outcome mode construction
  Sample K complete rollouts $\{\tau^{(k)}\}_{k=1}^{K}$
  Extract final answers $\{y^{(k)}\}_{k=1}^{K}$ and cluster them into semantic outcome modes $\mathcal{C}(q)$
  // Phase 2: Build a reliability-calibrated target over latent future states
  for each semantic mode $c \in \mathcal{C}(q)$ **do**
    Set $S_c = \{k : c(k) = c\}$ and empirical mass $m(c \mid q) = |S_c|/K$
    Select representative references $R(c) \subseteq c$ and normalized weights $\{\alpha_y^{(c)}\}_{y \in R(c)}$
    Estimate mode reliability $r(c, q)$ from semantic or evidence support of $R(c)$
  end for
  Form the reliability-calibrated target over latent future states $q_\theta(c \mid q) \propto m(c \mid q) \exp(\eta r(c, q))$ for all $c \in \mathcal{C}(q)$
  for each rollout $\tau^{(k)}$ **do**
    // Phase 3: Score each turn by the change in support for its realized mode
    for turn $t = 1, \ldots, T_k$ **do**
      Score each $y \in R(c(k))$ under prefixes $\tau_{\le t}^{(k)}$ and $\tau_{\le t-1}^{(k)}$ using the frozen copy
      // Multi-reference support approximates the full semantic cluster mass
      Aggregate scores into multi-reference cluster supports $\tilde{p}_\theta(c(k) \mid \tau_{\le t}^{(k)}) = \sum_{y \in R(c(k))} \alpha_y^{(c(k))} p_\theta(y \mid \tau_{\le t}^{(k)})$ $\tilde{p}_\theta(c(k) \mid \tau_{\le t-1}^{(k)}) = \sum_{y \in R(c(k))} \alpha_y^{(c(k))} p_\theta(y \mid \tau_{\le t-1}^{(k)})$
      Compute the outcome potentials $\Phi_\theta(\tau_{\le t}^{(k)}, c(k))$ and $\Phi_\theta(\tau_{\le t-1}^{(k)}, c(k))$
      Compute $r_{t,\mathrm{proc}}^{(k)} = \Phi_\theta(\tau_{\le t}^{(k)}, c(k)) - \Phi_\theta(\tau_{\le t-1}^{(k)}, c(k))$
      Set $\bar{r}_t^{(k)} = \lambda r_{t,\mathrm{proc}}^{(k)}$
    end for
    Add terminal outcome mass at the answer turn: $\bar{r}_{T_k}^{(k)} \leftarrow \bar{r}_{T_k}^{(k)} + (1 - \lambda) r_{\mathrm{final}}^{(k)}$
  end for
  // Phase 4: Convert augmented rewards into turn-level advantages
  Pool all $\{\bar{r}_t^{(k)}\}_{k,t}$ within the query group and z-normalize using the group's own mean μ_g and std σ_g to obtain $\{\tilde{r}_t^{(k)}\}_{k,t}$
  for each rollout $\tau^{(k)}$ **do**
    Compute turn advantages $A_t^{(k)} = \sum_{u=t}^{T_k} \gamma^{u-t} \tilde{r}_u^{(k)}$
    Assign $A_t^{(k)}$ only to trainable policy tokens generated in turn t
  end for
end for
// Phase 5: Optimize only policy-generated tokens with turn-specific credit
Update π_θ with the SIOP objective

D.2 Factorization and Supervised Limit

Proposition 2 (Factorization of outcome-potential shaping)**.** *The per-turn SIOP reward induced by Φ_θ factors as*

$$r_{t,\mathrm{proc}}^{(k)} = \underbrace{q_\theta(c(k) \mid q)}_{\text{calibrated target mass}} \cdot \underbrace{\Big[\log\big(\tilde{p}_\theta(c(k) \mid \tau_{\le t}^{(k)})\big) - \log\big(\tilde{p}_\theta(c(k) \mid \tau_{\le t-1}^{(k)})\big)\Big]}_{\text{cluster-level support gain}}, \tag{25}$$

i.e., cluster-level support shaping weighted by the calibrated target mass of that cluster. In the supervised limit where $q_\theta(c^\star \mid q) = 1$, $R(c^\star) = \{y^\star\}$, and $\alpha_{y^\star}^{(c^\star)} = 1$, every rollout with $c(k) = c^\star$ receives the standard gold-answer support shaping reward

$$r_{t,\mathrm{proc}}^{(k)} = \log p_\theta(y^\star \mid \tau_{\le t}^{(k)}) - \log p_\theta(y^\star \mid \tau_{\le t-1}^{(k)}), \tag{26}$$

while every rollout with $c(k) \neq c^\star$ receives $r^{(k)}_{t,\mathrm{proc}} = 0$ at every turn.

Proof. **Factorization.** The factored form follows by expanding the difference of two outcome potentials in Section 4. The calibrated target-mass factor $q_\theta(c(k) \mid q)$ is query-specific and constant over turns of the same rollout, so it can be grouped outside the log-difference of the multi-reference cluster support. The remaining factor is the turn-wise change in log cluster support for the realized cluster $c(k)$. Together they give "target mass times cluster-level support gain," which is the stated factorization.

Supervised limit. When a gold answer $y^\star$ is available and the self-induced target distribution concentrates on the gold cluster $c^\star$, the assumptions $q_\theta(c^\star \mid q) = 1$, $R(c^\star) = \{y^\star\}$, and $\alpha^{(c^\star)}_{y^\star} = 1$ imply $\tilde{p}_\theta(c^\star \mid \tau^{(k)}_{\leq t}) = p_\theta(y^\star \mid \tau^{(k)}_{\leq t})$. For a rollout with $c(k) = c^\star$, substituting into the factored reward collapses the target mass to 1 and the cluster support to the gold-answer likelihood, yielding

$$r^{(k)}_{t,\mathrm{proc}} = \log p_\theta(y^\star \mid \tau^{(k)}_{\leq t}) - \log p_\theta(y^\star \mid \tau^{(k)}_{\leq t-1}), \tag{27}$$

which is the standard supervised log-support reward used by turn-level shaping methods. For a rollout with $c(k) \neq c^\star$, $q_\theta(c(k) \mid q) = 0$, so the factored reward vanishes at every turn. The equivalence to gold-answer shaping therefore holds on rollouts that reach the gold cluster. The remaining rollouts receive no turn-specific credit and are learned from only through the group-relative comparison with gold-reaching rollouts, as in outcome-level training.

D.3 Bayesian View and Reliability Calibration

The self-induced target distribution can be read as a calibrated belief over semantic outcome modes obtained by a Bayesian-style update. The empirical mass $m(c \mid q)$ is the rollout-induced prior over latent future states, while the factor $\exp(u_\theta(c, q))$ occupies the algebraic position of an evidence likelihood derived from generic reliability signals. Under this view,

$$q_\theta(c \mid q) \propto \underbrace{m(c \mid q)}_{\text{rollout-induced prior}} \quad \underbrace{\exp(u_\theta(c, q))}_{\text{reliability likelihood proxy}}, \tag{28}$$

where the reliability strength is already absorbed into $u_\theta(c, q) = \eta r(c, q)$. This interpretation is not a claim that the NLI score is a calibrated likelihood. It only states that SIOP uses reliability signals in the same algebraic position as an evidence term before normalizing over semantic outcome modes.

Proposition 3 (Reliability calibration can reverse a spurious majority)**.** *Consider two semantic outcome modes c_s and c_r with empirical masses $m_s > m_r > 0$, where c_s is the higher-frequency mode and c_r is the lower-frequency mode. Let $u_\theta(c, q) = \eta r(c, q)$ with $\eta > 0$. The calibrated target assigns larger mass to c_r than to c_s if and only if*

$$r(c_r, q) - r(c_s, q) > \frac{1}{\eta} \log \frac{m_s}{m_r}. \tag{29}$$

Thus a lower-frequency mode can overtake a raw majority exactly when its reliability advantage exceeds the log frequency disadvantage scaled by $1/\eta$.

Proof. The normalized denominator of $q_\theta(\cdot \mid q)$ is shared by both modes, so $q_\theta(c_r \mid q) > q_\theta(c_s \mid q)$ is equivalent to

$$m_r \exp(\eta r(c_r, q)) > m_s \exp(\eta r(c_s, q)). \tag{30}$$

Taking logarithms and rearranging gives

$$\eta\big(r(c_r, q) - r(c_s, q)\big) > \log \frac{m_s}{m_r}, \tag{31}$$

which is equivalent to the stated condition. If the inequality is reversed, the higher-frequency mode remains preferred. Equality gives tied calibrated mass. □

The right-hand side $(1/\eta)\log(m_s/m_r)$ is the frequency disadvantage of c_r measured in the units of the chosen reliability score. Larger η makes reliability evidence sharper and lowers the threshold a low-frequency mode must clear. This condition should not be read as a guarantee that every wrong majority is corrected. If $r(c, q)$ is a bounded NLI confidence, then highly imbalanced majorities may be impossible to reverse under a fixed calibration strength; with $K{=}4$ and $\eta{=}1$, for example, a 3:1 majority cannot be reversed (Appendix F). The proposition instead identifies the exact regime in which reliability evidence can overcome raw self-consistency.

Case Study: SIOP on a multi-hop query

Query: When is the director of film Pudhu Vazhvu's birthday?
Ground-truth: 1 March 1910
SIOP rollout:
<think> To find the birthday of the director of "Pudhu Vazhvu", I first need to identify the director. Let me search for the director of this film. </think>
<tool_call> director of Pudhu Vazhvu film </tool_call>
<tool_response> Doc 1 (Title: "K. S. Ravikumar"): K. S. Ravikumar (born 30 May 1958) is an Indian film director and actor, primarily working in Tamil cinema... Doc 3 (Title: "Pudhu Vazhvu"): Pudhu Vazhvu is a 1957 Indian Tamil language film directed by M. K. Thyagaraja Bhagavathar. </tool_response>
<think> Doc 3 confirms that "Pudhu Vazhvu" was directed by M. K. Thyagaraja Bhagavathar. The retrieved documents do not give his birthday, so I need a second query for that. </think>
<tool_call> birthday M. K. Thyagaraja Bhagavathar </tool_call>
<tool_response> Doc 1 (Title: "M. K. Thyagaraja Bhagavathar"): Mayavaram Krishnamurthy Thyagaraja Bhagavathar (1 March 1910 to 1 November 1959), also called M.K.T., was an Indian actor, producer and Carnatic music singer... </tool_response>
<think> The retrieved document states his birth date as 1 March 1910. </think>
<answer> 1 March 1910 </answer>

Figure 4: Case study of a SIOP rollout on a multi-hop question. The agent issues two search queries: the first to identify the director, the second to retrieve his birthday. The final answer matches the gold label.

Table 6: Per-step wall-clock time for the two phases where SIOP and GRPO differ materially (mean over the last five steps). Rollout generation, reference-model forward passes, and actor updates are similar within measurement noise and omitted.

Phase	SIOP	GRPO	Δ
Scoring + NLI (per step)	22.4 s	0.00 s	+22.4 s
Advantage (per step)	6.3 s	0.03 s	+6.3 s
Total (per step)	141.0 s	115.0 s	+26.0 s (+23%)
Wall-clock (300 steps, 1 node)	11.75 h	9.6 h	+2.1 h
GPU-hours (8×A800 per run)	94	77	+17

E Additional Results

E.1 Compute Overhead

We profile SIOP and GRPO under identical settings (Qwen3-4B, 8×A800-80GB, same retriever, same agent loop, $K=4$ rollouts, 300 optimizer steps) and report in Table 6 the two phases where the per-step wall-clock time differs materially. Rollout generation, reference-model forward passes, and the FSDP actor update are similar within measurement noise across the two methods. SIOP's overhead is concentrated in two components. The first is *scoring* (average 22.4 s per step), dominated by the pseudo-response forward pass that evaluates $\log p_\theta(y \mid \tau_{\le t})$ at every turn boundary. The second is *advantage computation* (average 6.3 s per step), where turn-level per-query-group z-normalization and backward discounted accumulation across turn boundaries replace the single vectorized broadcast used by GRPO. Aggregated across all 300 steps, SIOP takes 11.75 wall-clock hours on 8×A800-80GB (94 GPU-hours per run), while GRPO takes 9.6 wall-clock hours (77 GPU-hours per run). SIOP therefore incurs an additional 2.1 hours and 17 GPU-hours per training run for verifier-free turn-level reward computation.

E.2 Additional Backbones and Tasks

Additional model family. We train SIOP on Phi-4-mini-instruct (3.8B) (Microsoft et al., 2025) for 300 steps in the same search environment. Table 7 shows gains on all seven benchmarks: average

Table 7: SIOP on Phi-4-mini-instruct (3.8B) after 300 training steps.

Model	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	Avg.
Exact Match								
Phi-4-mini	0.111	0.445	0.256	0.180	0.123	0.057	0.296	0.210
+ SIOP	0.275	0.580	0.451	0.375	0.369	0.143	0.376	0.367
Gain	+0.164	+0.135	+0.195	+0.195	+0.246	+0.086	+0.080	+0.157
Token-level F1								
Phi-4-mini	0.261	0.528	0.300	0.265	0.178	0.128	0.386	0.292
+ SIOP	0.419	0.664	0.490	0.469	0.432	0.254	0.458	0.455
Gain	+0.158	+0.136	+0.190	+0.204	+0.254	+0.126	+0.072	+0.163

Table 8: Multi-turn tool-integrated reasoning on MATH500. All methods start from Qwen3-8B and train for 100 steps.

Method	Training signal	MATH500
Qwen3-8B (base)	none	0.453
ZeroTIR-8B	gold-answer reward	0.825
SimpleTIR-8B	gold-answer reward	0.853
SIOP-8B	no answer labels	0.801

EM rises from 0.210 to 0.367 and F1 from 0.292 to 0.455, so the self-induced signal is not specific to the Qwen family.

Additional agentic task. We also evaluate multi-turn tool-integrated reasoning (TIR) on MATH500 (Lightman et al., 2024), where the agent alternates between reasoning and Python execution before producing a final answer. This changes both the task domain and the tool while keeping the sequential credit-assignment problem. All methods start from Qwen3-8B and train for 100 steps. Without answer labels, SIOP raises accuracy from 0.453 to 0.801, within 2.4 points of ZeroTIR (Mai et al., 2025) and 5.2 points of SimpleTIR (Xue et al., 2026), both trained with gold-answer rewards (Table 8).

E.3 Semantic Modes and Target Quality

Semantic diversity. We sample 1,280 queries with four rollouts each and cluster all extractable answers with the query-conditioned bidirectional-entailment criterion. Paraphrases such as "25 May 1965" and "May 25, 1965" are merged before counting, so multiple modes indicate semantically non-equivalent answers. Table 9 shows that the base Qwen3-4B policy produces multiple modes for 42.4% of queries with an extractable answer, compared with 20.8% after SIOP training, consistent with the trained policy concentrating on fewer semantic modes.

Alignment of the target with gold answers. This analysis tests the premise of the reward design: modes that receive larger target mass q_θ should be more likely to be correct, even though gold answers are never used in training. For the SIOP-trained Qwen3-4B and Qwen3-8B checkpoints, we sample 1,280 queries with four rollouts each, form semantic modes and compute q_θ as in training, and only then compare each mode with the gold answers. A mode is labeled *correct* if any of its answers and any gold answer entail each other given the query; as an NLI-free alternative, it is labeled correct if any of its answers reaches a token F1 of at least 0.8 with any gold answer. Table 10 reports four metrics. (i) *Cluster-level AUC*: pooling the modes of all queries, the probability that a randomly chosen correct mode receives a larger q_θ than a randomly chosen incorrect one, where 0.5 corresponds to random ranking. (ii) The same AUC under the token-F1 label. (iii) *Top-mode token F1*: for the mode with the largest q_θ in each query, the highest token F1 between any of its answers and any gold answer, averaged over queries. (iv) *Top-mode gold-equivalence rate*: the fraction of queries whose largest-q_θ mode is correct under the entailment label. The AUC is 0.667 (4B) and 0.733 (8B) under the entailment label and 0.663 and 0.726 under the token-F1 label, so the result does not hinge on using an NLI model to define correctness. The largest-q_θ mode reaches a token F1 of 0.500 and 0.477 and matches a gold answer for 44.7% and 44.4% of queries. Thus q_θ ranks semantic modes by their likelihood of correctness without gold supervision.

Table 9: Semantic-mode statistics over 1,280 queries with four rollouts each, restricted to queries with at least one extractable answer. Pairwise disagreement is the probability that two valid rollouts for the same query fall into different modes, averaged over queries.

Policy	One mode	Multiple modes	Modes per query	Pairwise disagreement
Qwen3-4B	57.6%	42.4%	1.599	34.4%
Qwen3-4B + SIOP	79.2%	20.8%	1.244	14.0%

Table 10: Agreement between the self-induced target q_θ and gold answers, which are not used during training. AUC ranks modes by q_θ against a gold-match label, where 0.5 corresponds to random ranking. The top mode is the mode with the largest q_θ in each query.

Metric	Qwen3-4B	Qwen3-8B
AUC (entailment label)	0.667	0.733
AUC (token F1 ≥ 0.8 label)	0.663	0.726
Top mode: token F1	0.500	0.477
Top mode: gold-equivalence rate	0.447	0.444

E.4 Case Study

Qualitative rollout. Figure 4 gives a qualitative trace of a full SIOP rollout on the query *"When is the director of film Pudhu Vazhvu's birthday?"*. The trajectory requires two distinct pieces of evidence: the first search identifies the director as M. K. Thyagaraja Bhagavathar, and the second search retrieves his birth date. This example illustrates the basic credit-assignment problem. The final answer confirms that the trajectory succeeds, but it does not reveal how much each retrieval step contributed. SIOP instead scores each turn by whether it increases support for the rollout's semantic outcome mode, so the bridge-entity search and the birth-date search can receive different credit.

F Reward Hacking and Confident-but-Wrong Modes

In verifier-free optimization, self-generated targets can reward an answer mode simply because many rollouts converge to it. This creates a reward-hacking path for confident-but-wrong modes: semantic agreement may reflect repeated model bias rather than evidence-grounded correctness. SIOP addresses this by separating cluster frequency from reliability before either signal enters the reward. In the factorized structure of Proposition 2, the target distribution $q_\theta(c \mid q)$ is not a direct function of cluster frequency. It combines empirical mode mass $m(c \mid q)$ with a reliability correction $u_\theta(c, q)$ driven by evidence support. Proposition 3 makes this behavior explicit: a lower-frequency but evidence-consistent cluster can receive larger target mass whenever its reliability advantage is large enough to overcome the log frequency gap. Thus raw majority support does not automatically dominate the target distribution when competing modes exist.

The same separation carries into turn-level credit assignment. The outcome potential weights prefix support by the calibrated target mass of the rollout's semantic outcome mode, so turns that increase support for reliable modes receive stronger credit than turns that only make a high-frequency mode more predictable. Cluster-level averaging across multiple rollouts and references further prevents a single wording choice from dominating the target.

Scope of reliability calibration. Two limits follow directly from the definition of q_θ. First, if all rollouts fall into one mode, $q_\theta(c \mid q) = 1$ regardless of evidence, so calibration cannot down-weight a unanimous but wrong answer; after SIOP training, 79.2% of queries have a single mode (Appendix E.3). Second, with K=4, η=1, and $r(c, q) \in [0, 1]$, a 3:1 majority cannot be reversed, since reversal requires $r(c_{\text{minor}}, q) - r(c_{\text{major}}, q) > \ln 3 \approx 1.10$ (Proposition 3); a 2:1:1 split can be reversed when the evidence gap exceeds $\ln 2 \approx 0.69$, and a 2:2 tie whenever the evidence differs. In the default setting, calibration therefore acts mainly as soft reweighting: with full evidence for the minority mode and none for the majority, a 3:1 split moves from $0.75/0.25$ to about $0.52/0.48$. Guarding against unanimous errors would require an absolute reliability term, which we leave to future work.

G LIMITATIONS AND FUTURE WORK

Limitations. The evaluation covers search-augmented QA with Qwen3-4B, Qwen3-8B, and Phi-4-mini, and multi-turn tool-integrated reasoning on MATH500 with Qwen3-8B, settings where semantic answer modes and tool observations provide a controlled testbed for turn-level credit assignment. This scope keeps the evaluation focused, while leaving broader agentic environments to future study. Since SIOP uses generic semantic clustering and evidence-support models, its signal is most natural when final outcomes can be grouped by meaning and retrieved evidence can support or contrast those meanings. Settings with highly ambiguous outcomes, weak evidence, or fine-grained tool states may require adapting the outcome-state construction.

Future work. Future work could extend semantic outcome potentials in several directions. First, the same abstraction may apply beyond search-QA to agentic settings where trajectories end in semantically diverse outcomes, such as tool use, web navigation, and interactive code repair. Second, outcome states need not be limited to text-answer clusters. They could be defined over structured tool traces, multi-modal observations, or hierarchical semantic modes, allowing the reward to track richer forms of future progress. Third, the potential-based view opens the door to adaptive credit granularity, where the training signal is assigned not only at the turn level, but also to retrieved evidence, tool actions, or finer action spans when they are the main source of progress.