

Hallucination-Aware Multi-Agent RAG via Internal Attention Rewards

Runsong Jia¹ Qian Chang² Mengjia Wu¹
Ciprian Doru Giurcǎneanu² Jie Lu¹ Yi Zhang^{1*}

¹ University of Technology Sydney, Sydney, Australia

² University of Auckland, Auckland, New Zealand

{runsong.jia, mengjia.wu, jie.lu, yi.zhang}@uts.edu.au,
qcha783@aucklanduni.ac.nz, c.giurcaneanu@auckland.ac.nz

Abstract

Retrieval-augmented generation (RAG) grounds language models in retrieved evidence, but its outputs often contain claims unsupported by that evidence. Recent multi-agent RAG systems decompose the pipeline into cooperative decisions, yet they typically optimize all agents under a shared task reward such as EM or F1, which measures answer overlap rather than faithfulness to evidence. We propose a hallucination-aware multi-agent RAG framework that combines the task reward with an internal hallucination-risk signal computed from the generator’s attention-head norms. Three role-specific LoRA adapters, a query planner, an evidence selector, and an answer generator, share a frozen backbone. A lightweight critic is trained once on TruthfulQA truthful/false pairs and then kept frozen, providing a complementary risk score that is added to the shared reward during cooperative PPO training. Across three backbones (Qwen3-8B, Llama3-8B, Llama2-13B) and three benchmarks (RAGTruth, HotpotQA, 2WikiMultiHopQA), our method improves task performance and reduces hallucination rates over the strongest multi-agent baseline. Ablations and critic diagnostics show that the TruthfulQA-calibrated signal provides useful reward shaping that transfers across RAG tasks.

1 Introduction

Retrieval-augmented generation (RAG) grounds language models in external evidence rather than parametric memory (Lewis et al., 2020; Gao et al., 2024), with the goal of producing answers that are easier to verify. In practice, however, RAG systems still produce claims that the evidence does not support: an answer may name the correct entity while adding relations, dates, or numbers that the retrieved passages never mention (Ji et al., 2023).

These failures do not arise from a single stage of the pipeline. Ambiguous questions, noisy or partially relevant passages, and the generator itself can each lead to claims that go beyond the evidence (Niu et al., 2024), and a single end-to-end objective does not indicate which stage is responsible.

Multi-agent RAG responds to this multi-stage nature by decomposing the pipeline into cooperative, learnable stages (Chen et al., 2026). A query planner makes the information need explicit, an evidence selector decides what to condition on, and an answer generator produces the final response. With this decomposition, the key design choice is no longer the architecture but the reward. When the three agents are optimized under a shared task reward such as EM or F1, hitting the reference span is enough to obtain most of the reward, while additional claims that go beyond the evidence are barely penalized. The reward signal therefore shapes the agents toward answer overlap, not toward grounding each claim in the evidence.

This mismatch is illustrated in Figure 1: an answer can score well on task metrics while adding details that the evidence does not mention. Existing supervision signals for faithfulness mostly operate on the final surface form. Reference-based metrics compare the answer to a gold reference, and LLM-as-a-judge approaches evaluate the generated answer post-hoc. Both evaluate the final answer rather than the generation process, and both are difficult to use as a dense reward during reinforcement learning: reference-based metrics only score how much the answer matches a gold span, while LLM-as-a-judge requires an external call for every rollout. A signal computed from the generator’s own behavior during decoding would be cheaper to query and would capture the generation process itself, rather than only evaluating the final answer after it has been produced.

Recent interpretability work, however, suggests that the model already carries information about

*Corresponding author.

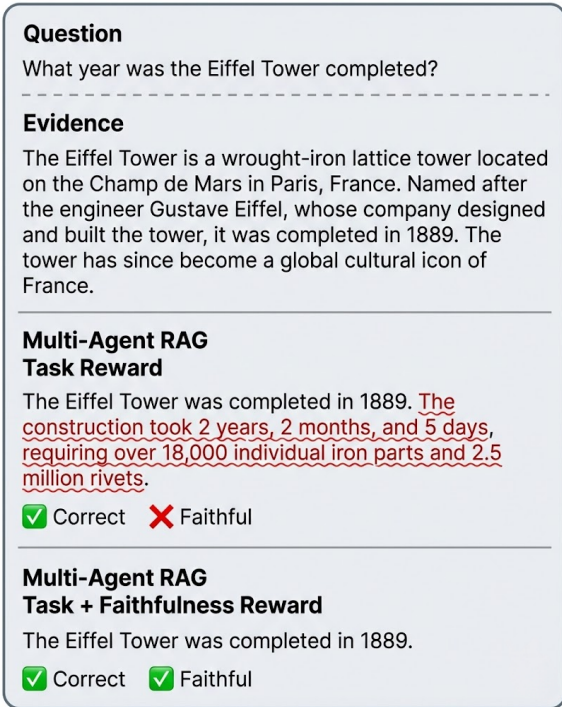


Figure 1: A task-reward-only model can answer the main question correctly while still adding details not supported by the evidence (wavy underline). This gap motivates a reward signal that targets faithfulness directly.

truthfulness in its internal representations (Burns et al., 2023; Li et al., 2023; Ho et al., 2025). In particular, simple statistics over attention-head outputs can separate truthful from false continuations, which makes them a practical source of signal during decoding. Existing methods use this kind of internal signal in two ways: to detect hallucinations after the answer has been produced (Su et al., 2024; Chen et al., 2024), or to steer activations during inference (Li et al., 2023). Both leave the policy unchanged and act only on the generator. In a multi-agent RAG pipeline, however, unsupported claims can also come from how the query is planned and which evidence is selected, so a faithfulness signal that only acts on the generator is not enough. This motivates our central question:

Can internal attention signals be turned into a reward that guides multi-agent RAG toward more faithful generation?

We answer this question with a hallucination-aware multi-agent RAG framework. The system decomposes RAG into three cooperative agents, a Query Planner, an Evidence Selector, and an Answer Generator, each implemented as a role-specific LoRA adapter (Hu et al., 2022) over a

shared, frozen backbone. A lightweight critic over attention-head norm features is trained once on TruthfulQA (Lin et al., 2022) truthful/false pairs and then kept frozen, providing a cheap internal faithfulness signal that can be queried at every roll-out. During cooperative PPO training, the critic’s risk score is combined with the task reward to form a shared training signal, so that all three agents are trained against both answer correctness and an internal risk estimate, and the faithfulness signal reaches the query planner and evidence selector as well as the generator.

Our main contributions are as follows:

- A hallucination-aware multi-agent RAG framework that jointly optimizes query planning, evidence selection, and answer generation under a shared reward combining task correctness with an internal faithfulness signal.
- A frozen risk critic over attention-head norm statistics, calibrated once on TruthfulQA and transferred across RAG benchmarks without retraining.
- Empirical evidence across three backbones and three QA benchmarks showing consistent improvements on task metrics and reductions in hallucination rate over the strongest multi-agent baseline.

2 Learning Setup

We present a hallucination-aware multi-agent RAG framework that uses internal attention behavior as a reward signal during reinforcement learning. The method is built around three design choices. First, we decompose RAG into query planning, evidence selection, and answer generation, so that each stage can be optimized as an explicit decision. Second, we keep the backbone language model frozen and learn role-specific LoRA adapters, which enables specialization while preserving the internal features used by the critic. Third, we train an offline attention-based critic and use its risk estimate as a reward-shaping term that complements the task reward.

Due to space constraints, the related work is discussed in Appendix A.1.

2.1 Problem Formulation

We study knowledge-intensive question answering in a retrieval-augmented setting. Each exam-

ple contains a question q and an evidence pool $D = \{d_1, \dots, d_K\}$, where D may be returned by a retriever or provided by the benchmark. The system outputs a natural-language answer a conditioned on the available evidence.

The learning objective has two parts: the answer should be correct with respect to a reference, and it should be faithful to the provided evidence. These goals are related but not identical—an answer can overlap with the reference while still containing unsupported claims (Niu et al., 2024; Ji et al., 2023)—and a terminal task reward alone does not explicitly distinguish them.

Let π_Θ denote the frozen backbone language model, parameterized by Θ . Each agent $u \in \{Q, E, G\}$ is instantiated with a fixed role prompt p_u and a role-specific LoRA adapter $\Delta\Theta_u$, which together define the agent policy

$$\pi_u(\cdot | x) = \pi_{\Theta + \Delta\Theta_u}(\cdot | p_u \oplus x), \quad (1)$$

where $\oplus$ denotes text concatenation. The prompts p_Q , p_E , and p_G specify the instructions for query planning, evidence selection, and answer generation, respectively.

2.2 Framework Overview

Figure 2 illustrates the overall framework. We formulate RAG as a turn-based cooperative decision process with three agents executed in a fixed order: Agent Q rewrites or plans the query, Agent E selects evidence from the candidate pool, and Agent G generates the final answer. The episode receives a single terminal reward shared by all agents.

The joint trajectory of one episode is $\tau = (q, \tilde{q}, D, \tilde{D}, a)$, where $\tilde{q}$ is the planned query, $\tilde{D} \subseteq D$ is the selected evidence subset, and a is the final answer.

Following the notation in Section 2.1, all agents are instantiated from the same frozen backbone π_Θ and differ only through their role-specific prompts and LoRA adapters. This design keeps the framework parameter-efficient while allowing each agent to specialize for its own role. Keeping the backbone and critic frozen during RL also preserves the feature space on which the critic was calibrated.

At the end of an episode, the frozen critic outputs a hallucination risk score $H \in [0, 1]$ from the generator-side answer representation. The shared terminal reward is

$$R = \alpha R_{\text{task}} - \lambda H, \quad (2)$$

where α and λ control the relative strength of task performance and hallucination-risk penalization.

2.3 Multi-Agent RAG Policies

The three agents share the same language-model interface but receive different prompts, observations, and LoRA adapters. This preserves a unified text-generation interface while assigning each policy a distinct functional role.

Agent Q: Query Planner. Agent Q converts the original question into a clarified query or lightweight plan, $\tilde{q} \sim \pi_Q(\cdot | q)$. The planned query makes salient entities, constraints, or sub-questions explicit. In our fixed-evidence setting, Agent Q does not trigger a new retrieval call, but provides a clearer conditioning signal for evidence selection and answer generation.

Agent E: Evidence Selector. Agent E chooses which evidence should condition the generator. This stage matters because irrelevant or redundant passages distract the generator, while missing key evidence forces it to rely on parametric memory.

Given q , $\tilde{q}$, and D , Agent E emits a textual selection of document identifiers, which is parsed into an evidence subset:

$$\tilde{D} \sim \pi_E(\cdot | q, \tilde{q}, D), \quad \tilde{D} = \{d_{i_1}, \dots, d_{i_M}\}. \quad (3)$$

We use a canonical textual schema for the selection action; invalid or duplicate identifiers are handled by the parser. This keeps evidence selection within the same language-model action interface as the other two agents.

Agent G: Answer Generator. Agent G is the only module that produces the free-form answer, so we define its policy over the question and selected evidence, and compute the hallucination-risk signal from its answer-side internal features: $a \sim \pi_G(\cdot | q, \tilde{D})$. Although Agent Q and Agent E do not generate the answer directly, they are trained under the same shared reward, so their intermediate decisions are shaped toward inputs that let the generator produce correct and faithful answers.

2.4 Attention-Based Hallucination Critic

The hallucination critic maps internal attention-head features to a scalar risk score. It is trained offline on a truthfulness calibration set and held frozen during policy optimization, providing a stable signal that guides the policy toward answer patterns associated with truthful continuations.

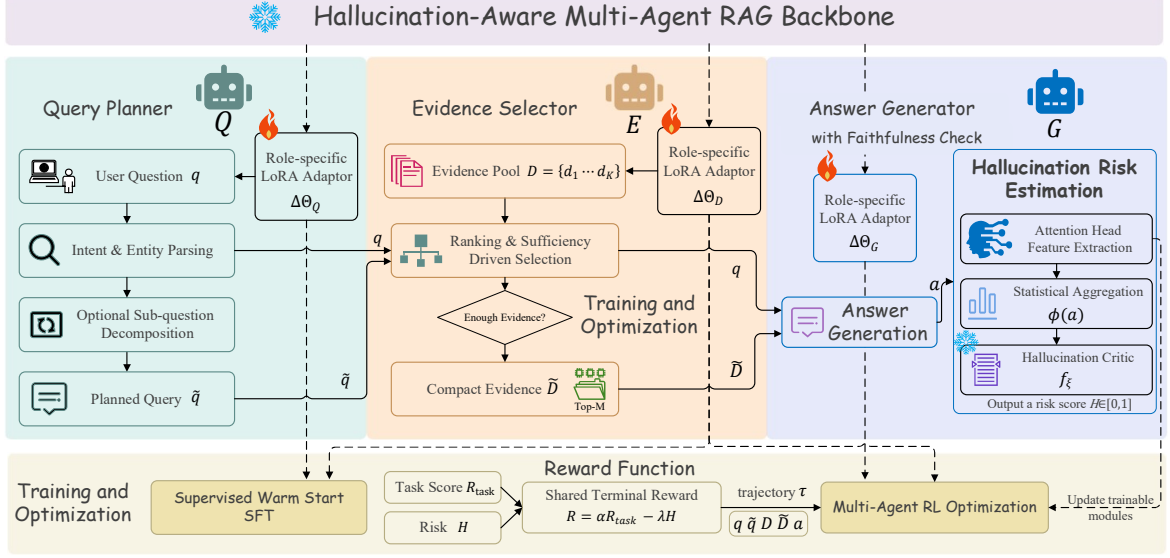


Figure 2: Overview of our hallucination-aware multi-agent RAG framework. The pipeline decomposes RAG into a query planner (Q), evidence selector (E), and answer generator (G), implemented as role-specific LoRA adapters on a frozen backbone. A frozen attention-based critic estimates a hallucination risk score from generator-side answer features; this score is combined with the task reward to form a shared terminal reward for cooperative PPO training.

Attention Heads as Truth Indicators. Prior work has shown that transformer internals carry information correlated with truthfulness (Burns et al., 2023; Li et al., 2023; Ho et al., 2025). In particular, NoVo (Ho et al., 2025) demonstrates that the norms of certain attention-head outputs can distinguish truthful from false answers on TruthfulQA. Building on this observation, we use attention head norms as compact internal features. Let $\mathbf{c}_t^{l,h}$ denote the output vector of head h in layer l at answer-token position t . We define the token-level head-norm feature as

$$x_t^{l,h} = \|\mathbf{c}_t^{l,h}\|_2. \quad (4)$$

This statistic is simple, model-internal, and available without an external hallucination annotator at RL time.

Head Feature Extraction. For each generated answer, we run a generator-side scoring pass and collect attention-head outputs over the answer tokens. Let $a = (a_1, \dots, a_T)$ denote the generated answer token sequence. For each selected head, we aggregate token-level head norms over the answer by mean pooling:

$$\bar{x}^{l,h}(a) = \frac{1}{T} \sum_{t=1}^T x_t^{l,h}. \quad (5)$$

The fixed-dimensional critic input is then

$$\phi(a) = \left[\bar{x}^{l,h}(a) \right]_{(l,h) \in \mathcal{H}_{\text{sel}}}, \quad (6)$$

where $\mathcal{H}_{\text{sel}}$ is the selected head set. Thus, $\phi(a)$ contains one pooled norm feature for each selected head. Additional implementation details are provided in Appendix A.2.2.

Truth-Related Head Selection. Not all heads carry useful truthfulness-related signal, so we select heads using a TruthfulQA-derived calibration set (Lin et al., 2022). Each calibration example contains a truthful answer and a false answer for the same question. For each head, we measure whether its pooled norm feature separates the two answers in a consistent direction.

For each truthful/false pair (s_i^+, s_i^-) , we compute

$$\Delta \bar{x}_i^{l,h} = \bar{x}_i^{l,h}(s_i^+) - \bar{x}_i^{l,h}(s_i^-). \quad (7)$$

We then define the pair-level vote as

$$v_i^{l,h} = \begin{cases} 1, & \text{sign}^{l,h} \cdot \Delta \bar{x}_i^{l,h} > 0, \\ 0, & \text{otherwise,} \end{cases} \quad (8)$$

where $\text{sign}^{l,h} \in \{+1, -1\}$ indicates the direction in which the head is correlated with truthful answers. The discrimination accuracy of each head is

computed as

$$\text{Acc}^{l,h} = \frac{1}{N} \sum_{i=1}^N v_i^{l,h}. \quad (9)$$

Heads exceeding an accuracy threshold are retained in $\mathcal{H}_{\text{sel}}$, yielding a compact set of truth-correlated heads as the critic’s input features.

Critic Architecture and Training. The critic is a lightweight MLP that maps the aggregated head features to a hallucination-risk probability:

$$H = f_{\xi}(\phi(a)) \in [0, 1]. \quad (10)$$

We train f_{ξ} with binary cross-entropy on the calibration pairs, assigning higher risk to false answers and lower risk to truthful answers. Once trained, the critic is fixed and used to provide the scalar risk term H during RL. Freezing both the critic and the backbone avoids a moving reward function during policy optimization and keeps the head-feature distribution stable across training.

2.5 Hallucination-Aware Multi-Agent RL

We integrate the critic into RL through a shared terminal reward. This keeps credit assignment simple: every agent receives the same scalar feedback, but each updates only its own role-specific adapter.

Supervised warm start. Before RL, we warm-start the three adapters with supervised instruction-following data, teaching the model to produce valid query plans, evidence selections, and answers under the corresponding prompts. RL then refines these role policies under the hallucination-aware reward. This separation keeps the exploration problem tractable and avoids relying on RL to discover the basic action formats from scratch.

Reward. After Agent G produces answer a , we compute the task reward as

$$R_{\text{task}} = \text{F1}(a, a^*), \quad (11)$$

where a^* is the reference answer. The frozen critic then estimates the hallucination risk $H = f_{\xi}(\phi(a))$ from the answer’s head-norm features. The final shared terminal reward follows Eq. 2, where the task reward is combined with the critic-based hallucination penalty. The coefficients α and λ are tuned on a development split.

PPO-Style Cooperative Multi-Agent Optimization.

We optimize the three agents with a cooperative PPO objective. All agents receive the same terminal reward defined in Eq. 2. We compute a shared advantage $A = R - b$, where b is an exponential-moving-average baseline of recent episode rewards. The advantage is batch-normalized when its variance is sufficiently large.

Each agent $i \in \{Q, E, G\}$ emits a textual action $z_i = (y_{i,1}, \dots, y_{i,T_i})$ under its role-specific observation o_i . Let $\theta_i = \Theta + \Delta\Theta_i$ denote the frozen backbone combined with the LoRA adapter of agent i . During the PPO update, we recompute the length-normalized log-likelihood of the sampled action:

$$\bar{\ell}_{\theta_i}(z_i | o_i) = \frac{1}{T_i} \sum_{t=1}^{T_i} \log \pi_{\theta_i}(y_{i,t} | o_i, y_{i,<t}). \quad (12)$$

Here, $y_{i,t}$ is the t -th token of agent i ’s textual action, and π_{θ_i} denotes the token-level distribution induced by that agent’s policy.

The sequence-level likelihood ratio is

$$r_i = \exp\left(\bar{\ell}_{\theta_i}(z_i | o_i) - \bar{\ell}_{\theta_i^{\text{old}}}(z_i | o_i)\right), \quad (13)$$

where θ_i^{old} denotes the rollout policy used to generate z_i . The clipped PPO loss for each agent is

$$\mathcal{L}_{\text{PPO}}^{(i)} = -\mathbb{E}[\min(r_i A, \text{clip}(r_i, 1 - \epsilon, 1 + \epsilon) A)]. \quad (14)$$

We also add a small sequence-level regularizer to discourage abrupt deviation from the rollout policy:

$$\mathcal{L}_{\text{reg}}^{(i)} = \left(\bar{\ell}_{\theta_i}(z_i | o_i) - \bar{\ell}_{\theta_i^{\text{old}}}(z_i | o_i)\right)^2. \quad (15)$$

The total objective sums over the three agents:

$$\mathcal{L} = \sum_{i \in \{Q, E, G\}} \left(\mathcal{L}_{\text{PPO}}^{(i)} + \rho \mathcal{L}_{\text{reg}}^{(i)}\right). \quad (16)$$

We update only the role-specific LoRA adapters $\{\Delta\Theta_Q, \Delta\Theta_E, \Delta\Theta_G\}$ with separate optimizers, while keeping both the backbone Θ and the critic f_{ξ} frozen.

This cooperative objective allows all agents to learn from the same final outcome: the query planner is rewarded when its plan improves downstream evidence use, the evidence selector when its selected evidence supports grounded answering, and the generator when its final answer is both correct and low-risk under the frozen critic.

3 Experiments

3.1 Experimental Setup

We evaluate our framework on three knowledge-intensive QA benchmarks: RAGTruth (Niu et al., 2024), HotpotQA (Yang et al., 2018), and 2WikiMultiHopQA (Ho et al., 2020). All methods are tested under a fixed-evidence setting, where the same evidence pool is provided to each system, so that the comparison focuses on policy learning rather than retriever variability. We instantiate our method with three backbones: Qwen3-8B, Llama3-8B, and Llama2-13B. The baselines include vanilla RAG variants, SelfRAG (Asai et al., 2024), and recent multi-agent RAG methods, including MAIN-RAG (Chang et al., 2025), RAG-DDR (Li et al., 2025), MA-RAG (Nguyen et al., 2025), and MMOA-RAG (Chen et al., 2026). We report Recall/EM/F1 on RAGTruth and Acc/EM/F1 on HotpotQA and 2WikiMultiHopQA, using R/A to denote Recall for RAGTruth and Accuracy for the two multi-hop QA datasets in averaged columns. We additionally report Answer Hallucination Rate (AHR) on RAGTruth to measure the fraction of generated answers containing claims unsupported by the provided evidence. Further details on datasets, implementation settings, baselines, and evaluation protocols are provided in Appendix A.2.

3.2 Main Results

Table 1 reports the main comparison. Our method achieves the strongest average performance for all three backbones, with the largest gains on the first metric (RAGTruth Recall or QA Accuracy) and EM. On Qwen3-8B, it improves over the strongest multi-agent baseline, MMOA-RAG, by +2.35 R/A, +0.95 EM, and +1.29 F1 on average. The gains are visible across all three datasets: +2.07 Recall on RAGTruth, +2.22 Accuracy and +2.34 EM on HotpotQA, and +2.76 Accuracy and +1.89 EM on 2WikiMultiHopQA.

On Llama3-8B and Llama2-13B, our method also leads on the average across all three metrics, with consistent gains on Recall/Accuracy and EM. These results indicate that the improvements from the hallucination-aware reward generalize across backbone architectures and pre-training data, rather than depending on a particular model family.

Compared with other multi-agent and reward-based methods, the difference lies in the reward signal rather than in pipeline structure. The internal hallucination-risk estimate provides supervi-

sion that surface-form task rewards do not capture, and the average gains in EM and F1, which are more sensitive to deviations from the reference are consistent with this interpretation.

3.3 Faithfulness Analysis

Table 2 reports Answer Hallucination Rate (AHR) on RAGTruth, which directly measures the fraction of responses containing unsupported claims. Our method achieves the lowest AHR across all three backbones, showing a consistent reduction in unsupported generation compared with the baselines. The improvement is most pronounced on Qwen3-8B, where the absolute reduction over MMOA-RAG reaches 10.74 points. The same pattern also holds on Llama3-8B and Llama2-13B, with absolute reductions of 5.62 and 3.34 points, respectively. These results indicate that the proposed framework improves not only task-level evaluation scores, but also the practical faithfulness of generated answers.

Overall, the AHR results indicate that the hallucination-aware critic provides a useful reward-shaping signal for improving faithfulness. The internal risk estimate encourages the model to move away from unsupported generations, while the improvements in Table 1 show that this does not come at the expense of task performance. Instead, faithfulness and task-level effectiveness improve in parallel, suggesting that the proposed reward design aligns well with the overall objective of grounded answer generation.

3.4 Ablation Studies

Table 3 and Figure 3 analyze the contributions of the hallucination critic and the role-specific agent updates on Qwen3-8B. For the agent-update ablations, the full query–evidence–generation pipeline is kept unchanged at inference time. The ablated agent still participates in the pipeline, but its LoRA adapter is frozen at the SFT initialization during PPO, while the other agents continue to update. This allows us to study the effect of RL on each role without changing the overall system interface.

Removing the hallucination critic leads to a clear performance drop: average R/A decreases by 3.31 points and average F1 decreases by 2.59 points. This directly tests whether the TruthfulQA-calibrated internal risk signal contributes information beyond the task reward alone, and the observed drop suggests that it provides useful guidance during policy optimization. The agent-update ablations show a similar pattern: freezing any role de-

Method	RAGTruth			HotpotQA			2WikiMultiHopQA			Average		
	Recall	EM	F1	Acc	EM	F1	Acc	EM	F1	R/A	EM	F1
<i>Qwen3-8B</i>												
Zeroshot w/ SFT	29.08	25.31	33.91	28.78	24.68	33.47	30.21	23.96	36.42	29.36	24.65	34.60
Vanilla RAG	31.99	24.62	34.67	34.94	16.91	38.84	34.09	25.42	36.56	33.67	22.32	36.69
Vanilla RAG w/ SFT	37.18	33.31	44.49	38.47	35.28	43.36	38.41	33.74	44.45	38.02	34.11	44.10
SelfRAG	34.42	31.77	42.93	37.32	36.39	38.68	32.35	28.73	43.04	34.70	32.30	41.55
MAIN-RAG	38.69	36.62	45.49	38.02	35.73	43.51	39.13	35.37	44.78	38.61	35.91	44.59
MA-RAG	41.02	37.24	43.31	40.58	37.92	45.12	40.15	36.84	45.83	40.58	37.33	44.75
RAG-DDR	47.20	36.65	46.26	46.49	38.45	47.18	47.25	42.55	49.83	46.98	39.22	47.76
MMOA-RAG	46.01	39.56	46.30	47.12	37.68	46.74	47.86	43.95	50.05	47.00	40.40	47.70
Ours	48.08	38.18	47.89	49.34	40.02	48.44	50.62	45.84	50.63	49.35	41.35	48.99
<i>Llama3-8B</i>												
Zeroshot w/ SFT	28.45	24.78	34.52	27.92	23.85	32.68	29.38	23.15	35.59	28.58	23.93	34.26
Vanilla RAG	31.24	23.89	33.94	34.11	16.08	25.01	33.26	24.59	35.73	32.87	21.52	31.56
Vanilla RAG w/ SFT	39.52	35.67	43.86	41.64	34.45	46.53	37.58	32.91	47.53	39.58	34.34	45.97
SelfRAG	33.79	31.14	40.37	38.49	32.56	42.03	31.52	27.87	42.21	34.60	30.52	41.54
MAIN-RAG	39.96	37.87	42.86	43.19	33.94	44.68	38.34	34.54	47.95	40.50	35.45	45.16
MA-RAG	40.32	35.93	43.91	48.78	35.78	46.46	49.18	44.76	51.13	46.09	38.82	47.17
RAG-DDR	46.42	39.42	47.66	49.95	36.69	48.57	49.02	44.92	51.76	48.46	40.34	49.33
MMOA-RAG	47.05	39.18	46.94	49.83	38.91	49.81	49.56	45.28	52.06	48.81	41.12	49.60
Ours	47.74	40.63	48.38	50.54	41.37	49.95	50.57	45.69	51.98	49.62	42.56	50.10
<i>Llama2-13B</i>												
Zeroshot w/ SFT	28.71	20.35	28.28	28.71	22.72	29.51	27.17	23.33	28.46	28.20	22.13	28.75
Vanilla RAG	30.48	23.66	32.60	31.98	16.95	30.32	31.11	16.67	24.17	31.19	19.09	29.03
Vanilla RAG w/ SFT	41.22	32.34	41.87	38.83	32.84	40.46	39.45	34.78	41.41	39.83	33.32	41.25
SelfRAG	30.46	27.81	33.97	30.36	29.43	33.98	32.39	29.74	34.08	31.07	28.99	34.01
MAIN-RAG	41.73	34.64	41.53	40.06	37.77	40.55	40.17	36.41	40.82	40.65	36.27	40.97
MA-RAG	41.09	37.84	43.58	42.65	38.65	43.33	41.05	36.57	43.80	41.60	37.69	43.57
RAG-DDR	42.19	38.19	44.33	42.77	38.56	44.44	41.89	37.79	44.63	42.28	38.18	44.47
MMOA-RAG	42.68	38.73	44.82	43.12	39.08	44.97	42.35	38.19	45.13	42.72	38.67	44.97
Ours	46.09	42.73	46.92	45.55	40.47	44.54	48.79	40.50	46.99	46.81	41.23	46.15

Table 1: Main results across three datasets and backbones. The first metric is Recall for RAGTruth and Accuracy for HotpotQA and 2WikiMultiHopQA; rightmost columns average the first metric (R/A), EM, and F1 across datasets. Best results in **bold**.

Method	Qwen3-8B	Llama3-8B	Llama2-13B
Zeroshot w/ SFT	76.35	74.62	73.81
Vanilla RAG	51.29	50.36	44.27
Vanilla RAG w/ SFT	53.06	51.62	52.37
SelfRAG	34.42	37.29	36.15
MAIN-RAG	40.31	42.11	41.02
RAG-DDR	32.46	34.02	35.68
MA-RAG	32.99	35.32	37.59
MMOA-RAG	31.98	30.27	31.03
Ours	21.24	24.65	27.69

Table 2: Answer Hallucination Rate (AHR, %; lower is better) on RAGTruth.

grades performance, with the generator update having the largest effect. This is consistent with the generator’s direct role in producing the final answer, while the smaller but consistent drops from freezing the query planner or evidence selector indicate that upstream decisions also contribute to the final generation quality.

3.5 Critic Diagnostics and Head Analysis

Calibration source and supporting evidence.

We calibrate the critic on TruthfulQA rather than on dataset-specific faithfulness annotations. TruthfulQA provides paired truthful and false answers outside the RAGTruth retrieval distribution, which allows the critic to be used as a transferable internal risk signal instead of a dataset-specific hallucination classifier. This choice also avoids training the reward model directly on the annotations used for faithfulness evaluation.

We evaluate the critic from two perspectives. On held-out TruthfulQA pairs, the best calibrated critic achieves 75.92% AUROC, indicating that the selected attention-head norm features contain a non-random truthfulness signal. When used for RAG policy optimization, the critic also produces risk scores in an actionable range, rather than saturating near zero or one. Together with the transfer diag-

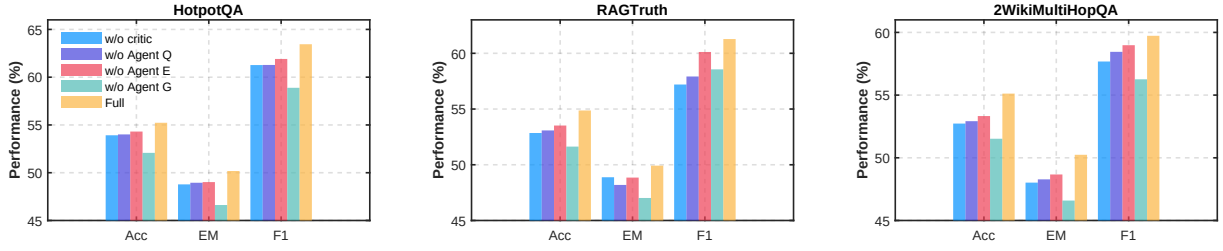


Figure 3: Ablation study of our framework (Qwen3-8B).

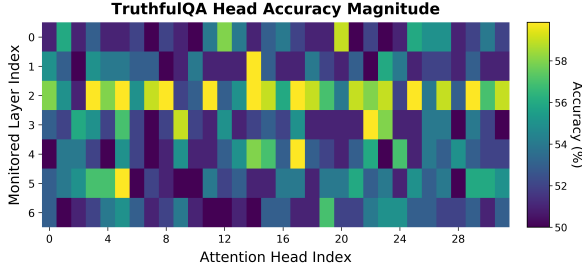


Figure 4: Head discrimination map on TruthfulQA. Each cell encodes the stronger of two head-level vote accuracies. Discriminative behavior is concentrated in a subset of heads spanning multiple layers.

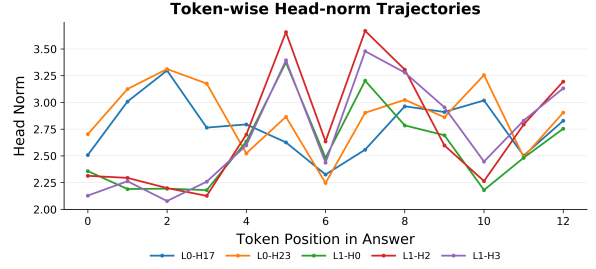


Figure 5: Token-wise head-norm trajectories for representative heads. Local changes appear near answer-salient tokens, indicating that head norms respond to content-bearing positions.

nostics in Appendix A.3.2, these results support our use of the critic as a lightweight auxiliary reward signal, rather than as a standalone hallucination detector.

We further examine which heads contribute to this signal and how their norms behave over generated answers.

Head discrimination map. Figure 4 visualizes the discrimination strength of individual attention heads on paired truthful and false answers. The signal is concentrated in a subset of heads across multiple layers, rather than being uniformly distributed throughout the model. This pattern supports the head-selection procedure in Section 2.4, which retains the heads with the strongest truth-correlated behavior as critic inputs.

Token-wise dynamics. Figure 5 shows token-wise head-norm trajectories for representative selected heads. The curves exhibit local changes around answer-salient tokens, such as entity mentions and factual predicates. This suggests that the selected head norms are sensitive to content-bearing positions in the answer, making them reasonable internal features for faithfulness-oriented reward shaping.

4 Conclusion

We presented a hallucination-aware multi-agent RAG framework that uses internal attention-head diagnostics as an auxiliary reward signal for reinforcement learning. The framework decomposes RAG into query planning, evidence selection, and answer generation, and trains role-specific LoRA adapters with a shared reward that combines task performance with a frozen critic’s hallucination-risk estimate. Experiments on RAGTruth, HotpotQA, and 2WikiMultiHopQA show consistent improvements across Qwen3-8B, Llama3-8B, and Llama2-13B, while also reducing answer hallucination on RAGTruth. These results suggest that lightweight model-internal signals can effectively shape RAG policies toward more faithful generation when used as a complementary reward rather than as a replacement for task supervision. More broadly, our findings indicate that the internal behavior of the generator can provide useful training-time feedback for grounded generation, beyond what is captured by answer-overlap metrics alone. We hope this encourages future work on combining external evidence supervision with model-internal diagnostics for more reliable RAG systems.

Limitations

Although our framework shows that model-internal signals can provide effective reward shaping for hallucination-aware multi-agent RAG, this work focuses on a lightweight and stable instantiation of this idea. First, the critic uses attention-head norm features as its internal signal, which keeps the reward inexpensive to compute during RL and allows the critic to remain simple and stable. Future work could examine whether richer diagnostics, such as hidden-state dynamics, MLP activations, or structured attention patterns, provide complementary faithfulness-related information. Second, we keep the critic frozen during policy optimization. This design preserves a fixed reward function and avoids changing the calibration of the internal features while the agents are being updated. Future work could explore adaptive critic-update strategies that track policy changes while preserving training stability.

Acknowledgments

Runsong Jia, Mengjia Wu, and Yi Zhang were supported by the Australian Commonwealth Scientific and Industrial Research Organization (CSIRO) in conjunction with the National Science Foundation (NSF) of the United States, under CSIRO-NSF #2303037. Mengjia Wu was also supported by Australian Research Council under Discovery Early Career Researcher Award DE260101493.

References

- Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pages 9112–9141.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.
- Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2023. [Discovering latent knowledge in language models without supervision](#). In *The Eleventh International Conference on Learning Representations (ICLR)*.
- Chia-Yuan Chang, Zhimeng Jiang, Vineeth Rakesh, Menghai Pan, Chin-Chia Michael Yeh, Guanchu Wang, Mingzhi Hu, Zhichao Xu, Yan Zheng, Ma-hashweta Das, and Na Zou. 2025. [MAIN-RAG: Multi-agent filtering retrieval-augmented generation](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2607–2622, Vienna, Austria. Association for Computational Linguistics.
- Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. Inside: LLMs’ internal states retain the power of hallucination detection. In *International Conference on Learning Representations*, volume 2024, pages 3056–3076.
- Yiqun Chen, Lingyong Yan, Weiwei Sun, Xinyu Ma, Yi Zhang, Shuaiqiang Wang, Dawei Yin, Yiming Yang, and Jiaxin Mao. 2026. Improving retrieval-augmented generation through multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 38:121336–121367.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and others. 2025. Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.
- Zhengyi Ho, Siyuan Liang, Sen Zhang, Yibing Zhan, and Dacheng Tao. 2025. [Novo: Norm voting off hallucinations with attention heads in large language models](#). In *International Conference on Learning Representations*, volume 2025, pages 88043–88066.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations (ICLR)*.
- Xiangkun Hu, Dongyu Ru, Lin Qiu, Qipeng Guo, Tianhang Zhang, Yang Xu, Yun Luo, Pengfei Liu, Yue Zhang, and Zheng Zhang. 2024. Knowledge-centric hallucination detection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6953–6975.
- Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 conference of the north american chapter*

- of the association for computational linguistics: Human language technologies (volume 1: Long papers)*, pages 7036–7050.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.
- Runsong Jia, Zhen Fang, Mengjia Wu, Jie Lu, and Yi Zhang. 2026a. Evidence-aligned entity verification for hallucination detection in retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 29538–29554.
- Runsong Jia, Mengjia Wu, Ying Ding, Jie Lu, and Yi Zhang. 2026b. Agent-enhanced heterogeneous graph rag for academic question answering. In *Proceedings of the ACM Web Conference 2026*, pages 8765–8768.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.
- Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in neural information processing systems*, 36:41451–41530.
- Xinze Li, Sen Mei, Zhenghao Liu, Yukun Yan, Shuo Wang, Shi Yu, Zheni Zeng, Hao Chen, Ge Yu, Zhiyuan Liu, Maosong Sun, and Chenyan Xiong. 2025. Rag-ddr: Optimizing retrieval-augmented generation using differentiable data rewards. In *International Conference on Learning Representations*, volume 2025, pages 9606–9627.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3214–3252.
- Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 5303–5315.
- Shengyu Mao, Yong Jiang, Boli Chen, Xiao Li, Peng Wang, Xinyu Wang, Pengjun Xie, Fei Huang, Hua-jun Chen, and Ningyu Zhang. 2024. Rafe: Ranking feedback improves query rewriting for rag. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 884–901.
- Thang Nguyen, Peter Chin, and Yu-Wing Tai. 2025. Ma-rag: Multi-agent retrieval-augmented generation via collaborative chain-of-thought reasoning. *arXiv preprint arXiv:2505.20096*.
- Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10862–10878.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Juntong Song, Xingguang Wang, Juno Zhu, Yuanhao Wu, Xuxin Cheng, Randy Zhong, and Cheng Niu. 2024. Rag-hat: A hallucination-aware tuning pipeline for llm in retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1548–1558.
- Weihang Su, Changyue Wang, Qingyao Ai, Yiran Hu, Zhijing Wu, Yujia Zhou, and Yiqun Liu. 2024. Unsupervised real-time hallucination detection based on the internal states of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14379–14391.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 10014–10037.
- Siheng Xiong, Oguzhan Gungordu, Blair Johnson, James C Kerce, and Faramarz Fekri. 2026. Scaling search-augmented llm reasoning via adaptive information control. *arXiv preprint arXiv:2602.01672*.
- Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Peixin Cao, Kaixin Ma, Jian Li, Hongwei Wang, and Dong Yu. 2024a. Chain-of-note: Enhancing robustness in retrieval-augmented language models. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 14672–14685.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024b. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *Advances in neural information processing systems*, 37:121156–121184.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. 2024. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131*.

A Appendix

A.1 Related Work

Our work draws on three lines of research: faithfulness in RAG systems, reinforcement learning for cooperative LLM agents, and internal signals for truthfulness. We situate our contribution by two questions: where the supervision signal can come from (external text or internal representations), and when it is applied (inference-time correction or training-time reward shaping).

A.1.1 Faithfulness and Multi-Agent Optimization in RAG

A growing body of work targets hallucination in RAG (Lewis et al., 2020; Gao et al., 2024; Ji et al., 2023; Niu et al., 2024; Jia et al., 2026a). Inference-time approaches augment the pipeline with self-critique or correction: Self-RAG (Asai et al., 2024) emits reflection tokens that trigger adaptive retrieval and self-evaluation, while CRAG (Yan et al., 2024) introduces a lightweight evaluator that detects insufficient retrieval and falls back on web search. RefChecker (Hu et al., 2024) contributes a fine-grained verification scheme by decomposing responses into claim triplets. On the training side, RAG-HAT (Song et al., 2024) demonstrates that DPO over faithful/unfaithful pairs can effectively reduce hallucinations, and RAFT (Zhang et al., 2024) shows how domain-specific RAG benefits from targeted fine-tuning.

A parallel direction decomposes RAG into cooperating modules and optimizes them jointly. Query rewriting (Ma et al., 2023; Mao et al., 2024), iterative retrieval-reasoning (Trivedi et al., 2023), and adaptive retrieval (Jeong et al., 2024) establish that individual stages benefit from being treated as learnable. Multi-agent formulations extend this idea: MMOA-RAG (Chen et al., 2026) casts RAG as cooperative MARL with shared F1-based rewards; RankRAG (Yu et al., 2024b) instruction-tunes a single model to perform both ranking and generation; and Chain-of-Note (Yu et al., 2024a) bridges retrieval and generation with explicit reading notes. Collectively, these methods demonstrate that decomposing RAG into learnable stages produces substantial gains over monolithic pipelines, and that text-level rewards such as F1 or preference judgments effectively optimize answer overlap. Our work builds directly on this line, adopting a similar multi-agent decomposition and PPO training framework, and extends it along a complementary axis: we examine whether a reward signal drawn from the model’s own internal computations can supplement these strong text-level objectives.

A.1.2 Reinforcement Learning for LLM Alignment

RL has become standard machinery for aligning LLMs. InstructGPT (Ouyang et al., 2022) established the SFT-then-PPO (Schulman et al., 2017) pipeline that underpins much of subsequent work; DPO (Rafailov et al., 2023) offered a streamlined alternative by directly optimizing over preference pairs. For reasoning, GRPO (Shao et al., 2024; Guo et al., 2025) shows that strong policies can be trained without a separate value network by estimating baselines from group statistics. Constitutional AI (Bai et al., 2022) opens up an additional dimension by deriving training signal from model-generated critique. In the RAG context, RAG-DDR (Li et al., 2025) contributes a differentiable data rewards approach for end-to-end training. And other works (Xiong et al., 2026; Jia et al., 2026b) show the ability of Agentic RAG and RL alignment. This line of work has converged on text-domain rewards: preferences, ratings, or match scores, which align language models with broad behavioral targets effectively. We retain the PPO machinery these methods established and explore a complementary source of supervision: signals computed from the policy’s own internal representations during generation.

A.1.3 Internal Signals for Truthfulness

A separate line of interpretability work has shown that transformer hidden states carry meaningful information about output truthfulness (Burns et al., 2023; Li et al., 2023; Ho et al., 2025; Su et al., 2024; Chen et al., 2024). Burns et al. (Burns et al., 2023) demonstrate that latent knowledge can be extracted from activations without supervision; ITI (Li et al., 2023) reveals that shifting attention head activations at inference time elicits more truthful responses; NoVo (Ho et al., 2025) achieves strong TruthfulQA accuracy through majority voting over attention-head norms; and MIND (Su et al., 2024) and INSIDE (Chen et al., 2024) show that internal states support effective post-hoc hallucination detection. These contributions reveal that internal model states carry actionable information about truthfulness, and they leverage this information either to detect hallucinations after generation or to steer activations during inference. Inspired by these findings, we explore a third application: using internal signals as a training-time reward, so that the information they expose can shape policy learning itself.

A.2 Experimental Details

A.2.1 Datasets Details

We evaluate on three knowledge-intensive QA benchmarks spanning single-hop and multi-hop reasoning tasks. **RAGTruth** (Niu et al., 2024), **HotpotQA** (Yang et al., 2018), and **2WikiMultiHopQA** (Ho et al., 2020). RAGTruth focuses on faithfulness in retrieval-augmented generation, while HotpotQA and 2WikiMultiHopQA require multi-hop reasoning over supporting evidence. We use a fixed-evidence setting in which all methods receive the same evidence pool. This controlled setup isolates the effect of policy learning from retriever variability. Below we introduce the datasets in more detail:

RAGTruth (Niu et al., 2024) is a hallucination-focused corpus containing nearly 18,000 response-level annotations from diverse LLMs in standard RAG settings. We use its QA subset, which provides question–evidence–answer triples with explicit retrieval contexts. Each instance includes word-level hallucination annotations, making it particularly suitable for evaluating faithfulness.

HotpotQA (Yang et al., 2018) contains 113K Wikipedia-based question-answer pairs requiring multi-hop reasoning over multiple supporting doc-

uments. Questions are diverse and not constrained to pre-existing knowledge schemas, with sentence-level supporting facts provided for explainability. We use the distractor setting where models must identify relevant information among 10 candidate paragraphs.

2WikiMultiHopQA (Ho et al., 2020) is a multi-hop dataset that combines structured (Wikidata) and unstructured (Wikipedia) sources. It contains approximately 190K examples with explicit reasoning paths expressed as entity-relation triples, enabling fine-grained evaluation of reasoning steps. Questions span four types: comparison, inference, compositional, and bridge-comparison.

To isolate our analysis from retriever quality and focus on the generation and reasoning components, we adopt a fixed-evidence setting where dataset-provided contexts serve as the evidence pool.

A.2.2 Implementation Details

We instantiate the framework with three backbones: Qwen3-8B, Llama3-8B, and Llama2-13B. The backbone weights stay frozen throughout training, and we train three role-specific LoRA adapters for query planning, evidence selection, and answer generation. Each adapter uses rank 16 and scaling factor 32 on the query, key, value, and output projections.

Training stages. Training proceeds in two stages. We first warm-start the three adapters with supervised fine-tuning for 1 epoch at learning rate 2×10^{-5} . We then run PPO for 1 epoch at learning rate 5×10^{-6} , with clip ratio $\epsilon = 0.2$, KL coefficient $\rho = 0.1$, task reward weight 1.0, and hallucination penalty weight $\lambda = 0.08$.

Hallucination critic. For each backbone, the critic reads attention-head norms from a fixed window of intermediate layers and keeps the truth-correlated heads above a 0.90 quantile threshold. Each selected head contributes four token-level statistics (mean, max, standard deviation, minimum), and a small MLP maps these features to a scalar risk score. The critic is trained with binary cross-entropy on TruthfulQA pairs and frozen thereafter.

A.2.3 Evaluation metrics.

For HotpotQA and 2WikiMultiHopQA, we report Accuracy (Acc), Exact Match (EM), and F1, following standard QA evaluation protocols. For RAGTruth, we report Recall, EM, and F1. We use

Recall instead of Accuracy as the first RAGTruth metric because our evaluation focuses on whether answer-relevant evidence tokens are recovered in the generated response. Accordingly, we use “R/A” in averaged columns to denote Recall for RAGTruth and Accuracy for HotpotQA and 2WikiMultiHopQA.

We additionally report **Answer Hallucination Rate (AHR)** on RAGTruth using GPT-4o-mini as an automated judge. AHR measures the fraction of generated answers that contain claims unsupported by the provided evidence:

$$\text{AHR} = \frac{|\{a \in \mathcal{A} : \text{contains_hallucination}(a, e)\}|}{|\mathcal{A}|}, \quad (17)$$

where $\mathcal{A}$ is the set of generated answers and e denotes the corresponding evidence. Unlike Recall, EM, or F1, AHR captures unsupported additions that may be missed by surface-form metrics when an answer contains the reference span but also introduces fabricated details.

A.2.4 Details about Baseline Models

We compare against methods spanning vanilla RAG, self-reflective approaches, and multi-agent systems:

- **Zeroshot w/ SFT**: The backbone LM fine-tuned on task data without retrieval augmentation.
- **Vanilla RAG**: Standard retrieve-then-generate pipeline using top- k retrieved passages as context.
- **Vanilla RAG w/ SFT**: Vanilla RAG with additional supervised fine-tuning on RAG-formatted examples.
- **SelfRAG** (Asai et al., 2024): A self-reflective framework that trains LMs to adaptively retrieve on-demand and critique generations using learned reflection tokens, enabling controllable trade-offs between retrieval frequency and output quality.
- **MAIN-RAG** (Chang et al., 2025): A multi-agent pipeline with independent query rewriting and answer generation modules, optimized separately via supervised learning.
- **RAG-DDR** (Li et al., 2025): A differentiable data rewards method that end-to-end trains RAG systems by aligning data preferences between retrieval and generation modules through rollout-based reward propagation.
- **MA-RAG** (Nguyen et al., 2025): A multi-agent RAG framework using collaborative chain-of-

thought reasoning with specialized agents for planning, extraction, and answer synthesis.

- **MMOA-RAG** (Chen et al., 2026): A multi-agent RL approach that jointly optimizes query rewriter, selector, and generator modules using MAPPO with F1-based task rewards, representing the closest baseline to our method but without faithfulness-aware signals.

A.3 Additional Results

A.3.1 Ablation Study

We provide comprehensive ablation analysis across all datasets and model architectures to validate each component’s contribution. Table 3 presents the complete ablation results on Qwen3-8B.

A.3.2 Additional Critic Transfer Diagnostics

This section provides additional diagnostics for the TruthfulQA-calibrated critic used in our hallucination-aware reward. The goal is not to train a task-specific hallucination classifier for each RAG benchmark, but to verify whether a critic calibrated on general truthful/false answer pairs can provide a useful auxiliary signal during downstream RAG policy optimization.

Truthfulness calibration. We first evaluate the critic on held-out TruthfulQA truthful/false answer pairs. After searching over the head-selection configurations, the best critic achieves **75.92% AUROC**, indicating that the selected head-norm features contain a substantive truthfulness signal. The critic is then kept frozen and used only as a scalar reward-shaping term during RAG policy optimization.

To examine whether the same internal features also capture evidence faithfulness, we train an evidence-aware critic using RAGTruth’s hallucination labels. Using the same attention-head-norm features, the critic achieves 74.05% AUROC on held-out RAGTruth data. This result shows that the internal features contain information related to evidence support in addition to general truthfulness.

We also examine whether the critic score is driven by simple surface properties. Its correlation with answer length is -0.1103 , while its correlation with answer-evidence word overlap is -0.1077 . Both values are close to zero, suggesting that the critic score is not simply a proxy for answer length or lexical overlap.

Cross Judge Robustness To examine whether the Answer Hallucination Rate (AHR) depends on

Variant	RAGTruth			HotpotQA			2WikiMultiHopQA			Average		
	Recall	EM	F1	Acc	EM	F1	Acc	EM	F1	R/A	EM	F1
w/o critic	43.82	35.96	44.42	47.26	36.75	45.96	47.04	43.35	48.83	46.04	38.69	46.40
w/o Q update	44.70	36.84	45.46	47.19	38.41	47.02	48.93	44.20	49.87	46.94	39.82	47.45
w/o E update	47.18	37.31	46.11	47.72	38.89	47.63	49.45	44.67	50.48	48.12	40.29	48.07
w/o G update	43.91	35.32	43.46	45.38	34.62	44.02	46.06	41.58	46.91	45.12	37.17	44.80
Full	48.08	38.18	47.89	49.34	40.02	48.44	50.62	45.84	50.63	49.35	41.35	48.99

Table 3: Ablation results with Qwen3-8B. In agent-update ablations, the named agent remains in the Q-E-G inference pipeline but is not updated during PPO. The average R/A column follows Table 1: Recall for RAGTruth and Accuracy for HotpotQA and 2WikiMultiHopQA.

Backbone	GPT-4o-mini	Qwen2.5-7B
Qwen3-8B	21.24	21.11
Llama2-13B	27.69	24.12

Table 4: Answer Hallucination Rate (%), lower is better) evaluated using two independent automatic judges.

a particular automatic judge, we repeat the evaluation using Qwen2.5-7B, a model from a different family from GPT-4o-mini. Table 4 compares the AHR scores produced by the two judges on two backbones.

The two judges produce similar AHR scores on both backbones, with the overall trend remaining consistent across model families. This suggests that the reported hallucination rates are not specific to GPT-4o-mini.

Shuffled-score control. To test whether downstream gains come from the sample-level critic signal rather than from the marginal distribution of penalty values, we run a shuffled-score control. In this control, the same frozen critic computes a risk score H for each rollout, but the scores are randomly reassigned across samples within each PPO update batch before computing the reward:

$$R = \alpha R_{\text{task}} - \lambda H_{\text{shuffled}}. \quad (18)$$

This preserves the scale and distribution of critic penalties while breaking their correspondence with the specific generated answers. If the critic were merely acting as a generic regularizer, this control would be expected to perform similarly to the calibrated critic. Instead, as shown in Table 5, the calibrated critic outperforms the shuffled-score control, suggesting that the sample-level risk signal is informative.

Reward-weight calibration. We also vary the hallucination penalty weight λ to examine the trade-off between the task reward and the internal risk

Setting	λ	Recall	EM	F1
w/o critic	0.00	43.82	35.96	44.42
shuffled-score critic	0.08	38.20	37.50	46.21
critic	0.04	47.98	35.03	46.96
critic	0.08	48.08	38.18	47.89

Table 5: Additional RAGTruth diagnostics for the TruthfulQA-calibrated critic. The shuffled-score control preserves the marginal distribution of critic penalties but breaks their sample-level correspondence with rollouts. The results show that the critic signal is useful as reward shaping, while the penalty weight affects the balance between the task reward and the internal risk penalty.

penalty. Among the tested settings, $\lambda = 0.08$ gives the strongest RAGTruth Recall, EM, and F1, while the shuffled-score control performs worse than the calibrated critic. This suggests that both the scale of the penalty and its sample-level correspondence with generated answers matter.

These diagnostics support two conclusions. First, the critic captures a real internal truthfulness signal on TruthfulQA, as reflected by its held-out AU-ROC. Second, when transferred to RAG policy optimization, the signal is most useful as a calibrated reward-shaping term: preserving the sample-level critic scores matters, but the penalty weight must be balanced against the task reward. This matches the design of our method, which treats the critic as a weak complementary signal rather than a standalone judge of RAG hallucination.

A.4 Prompt Templates

Each agent receives a fixed system prompt and a structured user prompt filled at episode time. The templates are shared across all backbones and datasets. Figure 6 shows the complete prompts for all three agents and the LLM-as-a-judge faithfulness evaluator. For Qwen3-8B, agent prompts are wrapped with the model’s chat template and thinking mode is disabled (enable_thinking=False)

to prevent <think> output from interfering with answer parsing.

A.5 Case Study

Figure 7 shows two representative inference traces illustrating the three-agent pipeline and the hallucination critic. Example 1 is a multi-hop question from HotpotQA where Agent E successfully narrows the retrieved passages to the two relevant ones, enabling Agent G to produce a correct and grounded answer. Example 2 is a causality question from 2WikiMultiHopQA where the retrieved passages do not contain the required information; the critic correctly identifies the resulting output as problematic and applies a training penalty.

A.6 Usage Claim of AI assistants

We use LLM for grammar and spelling checks only, with prompt “Proofread the sentences”. All conceptual development, analysis, writing, and editing were carried out solely by the authors without LLM assistance.

Agent Q — Query Planner System: You are a professional assistant skilled at rewriting complex or unclear questions into simpler, more searchable sub-questions. The rewritten sub-questions must have logical connections and dependencies, without being overly repetitive in meaning. Avoid using vague demonstrative pronouns. User: Please rewrite or decompose the following question into sub-questions, making them easier to search for answers in a search engine. Original question: <i>{question}</i> Requirements: (1) Output one sub-question per line. (2) Avoid duplicate sub-questions. (3) Avoid vague demonstrative pronouns. (4) At most 4 sub-questions. Output the sub-questions only, without any additional explanation.
Agent E — Evidence Selector System: You are a helpful assistant. Your task is to select the most relevant documents from the candidate list that can help answer the question. Output only the document IDs. User: Question: <i>{question}</i> Candidate Documents: <i>{documents}</i> Select the documents most helpful for answering the question. Output format: Document0, Document2, Document5 (comma-separated IDs). Only output the selected document IDs, nothing else.
Agent G — Answer Generator System: You are a helpful, respectful and honest assistant. Your task is to predict the answer to the question based on the given documents. If you do not know the answer, please do not share false information. Answer the question as accurately and briefly as possible. User: Question: <i>{question}</i> Reference Documents: <i>{documents}</i> Based on the above documents, answer the question as briefly and accurately as possible. Only give the brief and accurate answer, nothing else.
LLM Judge — Faithfulness Evaluator System: You are an evidence-grounded evaluator. Given a Question, Evidence passages, and a Model Answer, determine whether the Model Answer is fully supported by the Evidence <i>only</i>. Use ONLY the provided Evidence; do not use external knowledge or common sense. Do not infer missing details. If any key claim is not directly supported or is contradicted, label UNSUPPORTED. Quote short verbatim spans from the passages you relied on. Return a JSON verdict with fields: label (SUPPORTED/UNSUPPORTED), issue_type, confidence, rationale, and evidence_spans. Input: Question: <i>{question}</i> Evidence: [passage_1] <i>{text}</i> ... Model Answer: <i>{answer}</i> Output JSON:

Figure 6: Prompt templates for the three RAG agents (Q, E, G) and the LLM-as-a-judge faithfulness evaluator. Slots in italics (*{question}*, *{documents}*, *{answer}*, *{text}*) are filled at inference time. Agent E additionally receives the planned sub-questions from Agent Q as a soft conditioning signal. The judge outputs a structured JSON verdict used in the Faithfulness Analysis (Section 3.3).

Example 1 — Correct pipeline (HotpotQA)
Question: <i>The football manager who recruited David Beckham managed Manchester United during what timeframe?</i> Agent Q: The question is well-formed; passed to retrieval unchanged. Agent E: Selects two relevant passages (Alex Ferguson’s biography and Manchester United’s managerial history) from 10 retrieved candidates; discards the remaining 8. Agent G: “1986 to 2013” (correct) Critic: Low hallucination score ⇒ reward unpenalised.
Example 2 — Critic detects problematic output (2WikiMultiHopQA)
Question: <i>Why did Grand Duke Kirill Vladimirovich of Russia’s wife die?</i> Agent E: Selects passages about the Grand Duke; none contain an explicit cause of death. Agent G: “not mentioned” (incorrect; gold: “stroke”) Critic: High hallucination score ⇒ hallucination penalty applied.

Figure 7: Two inference traces illustrating the pipeline and the hallucination critic. In Example 1, Agent E focuses retrieval to two relevant passages and Agent G produces a correct answer confirmed by a low critic score. In Example 2, the critic flags a refusal-style output caused by insufficient evidence, applying a training penalty.