SYNERGATE-RAG: LEARNING GRAPH-GROUNDED PROCESS REWARDS FOR MULTI-HOP QUESTION ANSWERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-hop retrieval-augmented generation requires a reader to connect facts across evidence sources, yet final-answer supervision provides little information about the validity of individual reasoning steps. A trajectory with a correct final answer can therefore receive full credit despite unsupported intermediate claims or disconnected evidence use. Knowledge graphs provide explicit factual constraints, but incomplete coverage limits their reliability as process supervision. We introduce SynerGate-RAG, which learns continuous graph utility from source matching, citation relevance, and validated contradiction witnesses. Unlike graph process reward models (PRMs) based on algorithmic correctness or implicit outcome signals, our framework learns explicit, source-grounded utility and controls its contribution to graph–text process rewards. The labels preserve partial support and separate unavailable evidence from contradiction. An evidence-conditioned PRM learns this utility, while a stepwise gate combines graph and text rewards under a source-availability mask. The resulting rewards guide reader training alongside answer correctness. Experiments on multi-hop question answering benchmarks show improved answer quality and support the complementary roles of reward learning and adaptive allocation. SynerGate-RAG thus turns incomplete structured knowledge into selective process supervision without reward-model calls at inference.

1 INTRODUCTION

Retrieval-augmented generation connects language models to external evidence (Lewis et al., 2020). Multi-hop questions require composing facts across entities and relations, yet answer correctness does not establish that this composition occurred. On HotpotQA, the DiRe probe attributed only 17.5 of XLNet's 71.9 answer-F1 points to multifact reasoning (Trivedi et al., 2020). This finding concerns a particular model and diagnostic setting, but exposes a general evaluation gap: a correct answer can conceal disconnected evidence use.

Outcome rewards have enabled reinforcement learning for retrieval-augmented reasoning (Jin et al., 2025; Song et al., 2025), but they cannot directly distinguish supported intermediate deductions from shortcuts that reach the same answer. Process supervision offers finer feedback (Lightman et al., 2024). In open-domain RAG, however, its reliability depends on the evidence available to assess each step. Passage-conditioned likelihood supplies broad feedback, but fluent continuations need not be factually justified. Graph relations make factual connections explicit, while their coverage is uneven.

A missing relation is not evidence of contradiction. In path-based KG reasoning, treating unobserved answers like false ones creates false negatives; soft reward shaping addresses this problem by crediting plausible unobserved answers (Lin et al., 2018). The corresponding risk in process supervision is to penalize a valid deduction because its supporting relation is absent from the retrieved graph. In our offline labeling, negative graph-utility targets require an explicit contradiction witness; missing evidence alone cannot create a negative label. Partial support should also remain distinguishable from full support.

Existing graph PRMs address related but different supervision problems. GraphPRM learns algorithmic step correctness (Peng et al., 2025), whereas DPRM derives implicit graph and textual process rewards from outcome supervision (Wang et al., 2026). These objectives leave open how

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.

incomplete factual evidence should supervise a reader and how strongly graph feedback should be weighted against text. We address these questions through source-grounded utility learning and availability-aware reward allocation.

SynerGate-RAG learns signed weak utility from source-backed graph checks on teacher trajectories. An evidence-conditioned PRM predicts this utility, while a separately fitted gate estimates graph-verdict strength. An availability mask excludes inadmissible graph feedback; the gate combines the remaining graph and text rewards at step boundaries. The scorers and gate stay frozen during reader optimization, and deployment requires no reward-model calls. Evaluation on HotpotQA, 2WikiMultiHopQA, and MuSiQue shows improved answer quality, with the largest gains on 2WikiMultiHopQA.

Our contributions are threefold:

- We formulate continuous graph utility as weak process supervision, preserving partial support and an explicit distinction between unavailable evidence and contradiction.
- We introduce an evidence-conditioned PRM and a stepwise gate that connect this supervision to graph–text reward allocation during reader training.
- We develop a controlled evaluation spanning answer quality, process-error identification, allocation utility, and graph incompleteness, examining how reward learning and allocation contribute to multi-hop answer quality.

2 Related Work

2.1 Retrieval-augmented multi-hop reasoning

IRCoT interleaves retrieval and reasoning (Trivedi et al., 2023); CoRAG learns retrieval chains (Wang et al., 2025), and ReaRAG incorporates knowledge-guided reasoning into iterative retrieval (Lee et al., 2025). FLARE triggers retrieval from low-confidence continuations (Jiang et al., 2023). These methods organize evidence acquisition and use. Self-RAG additionally learns reflection tokens to assess retrieval and generation (Asai et al., 2024). We instead learn source-backed graph utility for reader steps, holding evidence fixed during generation to isolate reward-design effects.

2.2 Knowledge graphs as evidence and supervision

GraphRAG uses community summaries (Edge et al., 2024), HippoRAG supports associative retrieval (Jiménez Gutiérrez et al., 2024), and KG^{2}RAG organizes retrieved chunks through factual relations (Zhu et al., 2025). RoG learns relation-path planning for graph-grounded retrieval and reasoning (Luo et al., 2024); MindMap combines graph evidence and implicit model knowledge through prompting (Wen et al., 2024). These approaches structure evidence and reasoning; we learn how incomplete evidence should contribute to process rewards. GraphPRM learns continuous scores from binary algorithmic-correctness labels (Peng et al., 2025), while DPRM learns implicit CoT and KG rewards from outcome preferences for inference-time selection (Wang et al., 2026). SynerGate-RAG learns explicit signed utility from source-backed checks and allocates graph–text feedback during reader training, without reward-model calls at inference.

2.3 Reinforcement learning and process supervision

Search-R1 and R1-Searcher optimize search-enabled reasoning with outcome rewards (Jin et al., 2025; Song et al., 2025). GraphRAG-R1 adds retrieval-efficiency constraints and cost-aware answer feedback (Yu et al., 2026); these signals do not directly assess intermediate factual support. ReasonRAG uses answer-guided process preferences for query, evidence, and answer actions (Zhang et al., 2025a), while Q-PRM learns query-rewriting preferences and retains PRM-guided decoding (Ye et al., 2025). We supervise reader steps against fixed source evidence and allocate graph–text feedback according to source availability. ProcessBench also shows why answer selection alone does not establish step-verification quality (Zheng et al., 2025), motivating our separate process evaluation. ArmoRM learns prompt-conditioned weights across response-level reward dimensions (Wang et al., 2024a); our gate instead estimates stepwise graph-verdict strength under an explicit availability mask.

The distinction lies in learning signed utility from incomplete factual evidence and controlling its contribution to reader training. Appendix F compares the supervision objects, targets, and uses of the closest process-supervision methods.

3 SYNERGATE-RAG

3.1 OVERALL FRAMEWORK

Let q be a question, D the retrieved passages, and G a source-traceable subgraph. A reader generates $y = (s_1, \ldots, s_T, a)$, where each step contains reasoning, cited triples, and an intermediate conclusion, followed by a final answer a. SynerGate-RAG consists of four connected modules: evidence and trajectory preparation, continuous graph-grounded reward learning, adaptive graph–text allocation, and stepwise policy optimization (Figure 1). Reference answers are reserved for training-side outcome supervision and evaluation; they are not inputs to evidence construction, PRM scoring, or the gate.

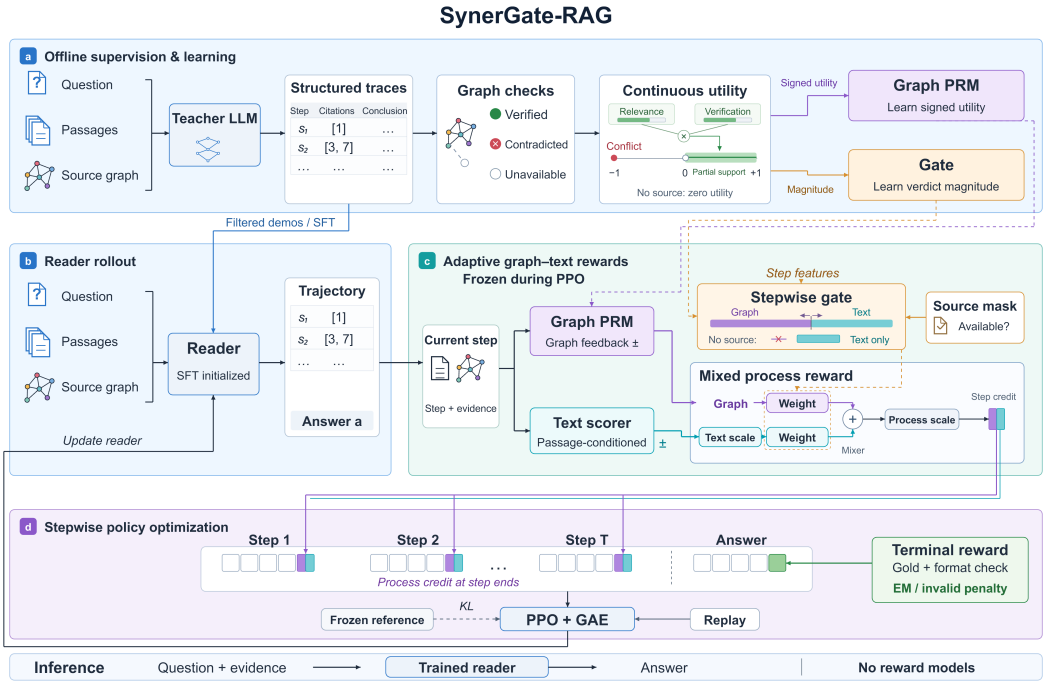

Figure 1: SynerGate-RAG. Offline graph supervision trains the continuous PRM and the gate. During reader optimization, frozen graph and text scorers supply step rewards through the frozen gate. Only the reader and its evidence preparation are required at inference.

The reader receives retrieved passages and a source-traceable subgraph (Figure 2), then generates intermediate conclusions, cites relevant supplied triples, and produces a final answer. An empty citation field declares that no graph triple is cited; it does not establish graph unavailability. Passages remain available. Evidence stays fixed throughout generation, without additional retrieval calls.

Evidence preparation ⟶ Reader generation ⟶ Answer extraction

Evidence-conditioned input

```
Question: <multi-hop question>

Retrieved Passages:
[1] Wikipedia Title: <page title>
<passage text>
[2] Wikipedia Title: <page title>
<passage text>
...

[Knowledge Graph Context]
[g1] (<A>, <r1>, <B>)
Qualifiers: <scope, if applicable>
[g2] (<B>, <r2>, <C>)
...
[End of Knowledge Graph]
```

Structured reader response

```
[Step 1]
Reasoning: <supported deduction>
Knowledge Used: [(<A>, <r1>, <B>)]
Conclusion: <sub-answer>

[Step 2]
Reasoning: <extend Step 1>
Knowledge Used: [(<B>, <r2>, <C>)]
Conclusion: <next conclusion>
...

[Final Answer]
<concise answer>
```

Knowledge Used copies supplied triples, or [] when none is cited. Each triple resolves to its visible qualifiers and cached source record (Appendix A). Reference answers are excluded.

Figure 2: Evidence-to-answer interface. The reader conditions each step on the supplied evidence and preceding steps, then emits a concise final answer. Angle brackets denote placeholders; this is a schema, not a measured model response.

3.2 Graph-guided trajectories and continuous supervision

Evidence preparation links entities in the question and passages to a bounded source-traceable subgraph; passage-derived navigation edges receive no graph credit. A teacher generates structured trajectories from this evidence, including questions with sparse graphs. Answer-level filtering selects reader demonstrations, whereas PRM construction also retains rejected or perturbed steps with interpretable evidence judgments. A correct final answer does not label every preceding step as correct.

Offline labeling combines deterministic source matching with a frozen, evidence-restricted teacher. Let $n_t = |C_t|$ count all declared citations, including unmatched ones. For citation i, V_{ti} indicates a source match and L_{ti} indicates relevance to the current reasoning or conclusion, with the citation field removed. Set $p_t = \sum_i V_{ti}/n_t$ and $v_t = \sum_i L_{ti}/n_t$; both are zero for empty citations. The teacher supplies relevance decisions and candidate contradiction witnesses. Deterministic checks validate witness references and conflict operations; $b_t = 1$ requires an accepted witness. A separate deterministic mask m_t tests source readability and entity–relation scope, without deciding semantic correctness. The weak target is

$$z_t = \begin{cases} -1, & b_t = 1, \\ p_t v_t, & b_t = 0,\ m_t = 1, \\ 0, & m_t = 0. \end{cases} \qquad z_t \in \{-1\} \cup [0, 1]. \tag{1}$$

We enforce $b_t \leq m_t$: missing evidence and rejected witnesses cannot create negative labels, and empty citations do not bypass contradiction checks. Zero denotes zero graph utility, not incorrect reasoning. The product $p_t v_t$ combines marginal match and relevance rates; it does not require the same citation to pass both checks and is not a correctness probability. Annotation failures receive no target, rather than zero utility. Appendix B specifies matching, witness validation, and failure handling. Semantic annotation is offline; PPO uses the deterministic mask with the frozen PRM and gate.

3.3 LEARNING AN EVIDENCE-CONDITIONED PROCESS REWARD

The graph PRM encodes $x_t = (q, D_t, G_t, s_{<t}, s_t)$ using fixed evidence $D_t = D$, $G_t = G$, selected without reference answers. Source records distinguish supplied facts from generated claims. Its scalar head predicts $r_t^G = \tanh(f_\psi(x_t))$, fitted to continuous utility with weighted Huber loss:

$$\mathcal{L}_{\text{PRM}} = \frac{\sum_t w_t \, \text{Huber}_\delta(r_t^G - z_t)}{\sum_t w_t}. \tag{2}$$

Weights w_t balance negative, zero, and positive targets using training counts (Appendix B); positive values remain continuous. Unavailable-source examples train zero utility, and the mask excludes their graph rewards regardless of prediction. Evidence-changing negatives and meaning-preserving paraphrases discourage format-only scoring. Learned negative predictions need not have validated contradiction witnesses: the PRM can inherit weak-label errors, so its step judgments are evaluated separately.

We repurpose the released ReaRAG-9B checkpoint (Lee et al., 2025) as a frozen text-likelihood scorer. We map the step's mean conditional negative log-likelihood ℓ_t, given the question, passages, and preceding steps, to $r_t^D \in [-1, 1]$ through a fixed monotone transformation (Appendix A). Graph inputs are omitted; this score measures conditional plausibility rather than entailment.

3.4 ADAPTIVE GRAPH–TEXT REWARD ALLOCATION

The gate models graph-verdict strength independently of its sign. We supervise the unmasked gate g_ω with $a_t = |z_t|$, allowing both supportive and contradictory evidence to influence reward allocation. Its features are graph density, entity-link resolvability, and step surprisal from a frozen reference reader (Appendix B):

$$u_t = [d(G_t), c_t, h_t], \quad g_\omega(u_t) = \sigma\left(\frac{W^\top u_t + b}{\tau}\right), \quad \alpha_t = m_t \, g_\omega(u_t). \tag{3}$$

$$\mathcal{L}_{\text{gate}} = -\mathbb{E}_t[a_t \log g_\omega(u_t) + (1 - a_t) \log(1 - g_\omega(u_t))]. \tag{4}$$

Density describes local evidence structure, link resolvability measures how step entities connect to that evidence, and surprisal measures deviation from the frozen reference reader. Surprisal is the realized step's mean negative log-probability, not semantic entropy. These features predict verdict strength, not whether graph feedback is superior to text. Inputs exclude reference answers and signed PRM scores. Gate fitting uses families disjoint from PRM fitting and frozen feature standardization; fixed, shuffled, and direct-verdict controls test its allocation utility.

The resulting process reward is

$$r_t^{\text{proc}} = \lambda_{\text{step}} \left[\alpha_t r_t^G + (1 - \alpha_t) c_D r_t^D\right]. \tag{5}$$

The coefficient c_D accounts for the different score scales. When source evidence is unavailable, $\alpha_t = 0$ and the process reward reduces exactly to textual feedback. For an admissible negative PRM score, increasing graph weight strengthens the graph penalty. Allocation changes the training reward while preserving the reader inputs.

3.5 STEPWISE POLICY OPTIMIZATION

The reader is initialized on coherent teacher demonstrations. For a generated trajectory, let $V(y)$ be a format-validity predicate that checks the step schema and the presence of a final answer, without inspecting answer correctness. The terminal reward combines answer correctness with an invalid-output penalty:

$$r^{\text{end}}(y) = \lambda_A[V(y) \, \text{EM}(a, a^\star) - (1 - V(y))]. \tag{6}$$

Among task-reward terms, trajectories with no parseable steps receive only the invalid-output penalty. Each process reward is assigned to the final token of its corresponding step; the terminal term is assigned at the response end. The reader is optimized with clipped PPO (Schulman et al., 2017) and token-level generalized advantage estimation (Schulman et al., 2016). A separate frozen copy of the initialized reader supplies the reference-policy KL penalty, and supervised replay anchors updates to the demonstrations. The trainable policy value head is distinct from the graph PRM. The PRM, text scorer, reference reader, and gate remain frozen throughout policy optimization.

At inference, the trained reader generates one response from an evidence bundle; evidence-preparation costs remain. Algorithm 1 summarizes training, with numerical settings, the output contract, and token-level updates in the appendices.

Algorithm 1 SynerGate-RAG Training

Require: Training pairs $(q, a^\star)$; retriever $\mathcal{R}$; source KG $\mathcal{K}$; teacher π_T; text scorer S_D.
Ensure: Trained reader π_θ.

Offline supervision preparation
Prepare evidence (D, G) for each q; generate teacher trajectories with π_T
Derive continuous graph utility z_t and availability m_t, Eq. (1)
Initialize π_θ on accepted demonstrations; $\pi_{\text{ref}} \leftarrow \text{Copy}(\pi_\theta)$
Train the graph PRM on z_t and gate on $|z_t|$, using disjoint families
Freeze $f_\psi, S_D, g_\omega, \pi_{\text{ref}}$ and normalization parameters
for each policy-training batch **do**
  $\pi_{\text{old}} \leftarrow \text{Copy}(\pi_\theta)$; sample responses $y \sim \pi_{\text{old}}(\cdot \mid q, D, G)$
  Stepwise reward evaluation
  for each response y in the batch **do**
    $\widehat{r} \leftarrow \mathbf{0}_{|y|}$; parse steps, their end tokens j_t, and final answer a
    for each parseable step s_t **do**
      Obtain scoring context x_t, availability m_t, and gate features u_t
      $r_t^G \leftarrow \tanh(f_\psi(x_t))$; compute text reward r_t^D with S_D
      $\alpha_t \leftarrow m_t g_\omega(u_t)$ ▷ $m_t = 0 \Rightarrow \alpha_t = 0$
      $\widehat{r}_{j_t} \leftarrow \widehat{r}_{j_t} + \lambda_{\text{step}}[\alpha_t r_t^G + (1-\alpha_t) c_D r_t^D]$ ▷ Step-end credit
    $\widehat{r}_{|y|} \leftarrow \widehat{r}_{|y|} + \lambda_A[V(y)\,\text{EM}(a, a^\star) - (1 - V(y))]$ ▷ Eq. (6)
  Policy update
  Apply KL against π_{ref} and compute GAE (Appendix B.5, Eqs. (7)–(8))
  Update reader θ and value head ν with clipped PPO on the batch
  if replay is scheduled **then** update θ on the accepted demonstrations
return π_θ

4 Experiments

4.1 Experimental setup

The evaluation design covers HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022). We report answer EM, token F1, and unweighted dataset macro averages. Table 1 summarizes three evaluation seeds (42, 43, 44); Table 2 repeats the full-method mean alongside the reported ablation scores; Tables 3–4 and Figure 3 report their respective point estimates. The controlled training comparisons fix passages, visible graphs, answer normalization, and decoding. All controlled variants start from the same reader initialization and use the same policy-training budget. Development selection and final reporting are separated by question and family. Dataset sizes and training settings appear in Appendix A.

The reference methods include Naive RAG, Self-RAG (Asai et al., 2024), R1-Searcher (Song et al., 2025), CoRAG (Wang et al., 2025), IRCoT (Trivedi et al., 2023), and ReaRAG (Lee et al., 2025). Table 1 compares reference systems; backbone, implementation, and retrieval differences can also affect its gaps. The shared-control conditions apply to Table 2, which compares training objectives. Process evaluation and candidate ranking use separate held-out families and exclude the gold-dependent outcome reward.

4.2 ANSWER QUALITY ACROSS DATASETS

Table 1: Answer quality (EM/F1, %). Mean over evaluation seeds 42, 43, and 44. Bold and underlining mark the best and second-best means.

Method	HotpotQA		2Wiki		MuSiQue		Macro	
	EM	F1	EM	F1	EM	F1	EM	F1
Zero-shot	0.1	4.1	0.1	4.2	0.1	3.0	0.1	3.8
Naive RAG	25.5	37.6	5.2	17.1	9.2	19.7	13.3	24.8
Self-RAG	17.2	28.9	10.5	24.4	8.8	20.4	12.2	24.6
R1-Searcher	<u>39.0</u>	<u>51.1</u>	<u>37.2</u>	44.9	17.3	26.9	31.2	41.0
CoRAG	20.2	28.6	18.4	29.0	12.6	18.0	17.1	25.2
IRCoT	37.5	49.5	35.1	46.5	22.2	34.3	<u>31.6</u>	43.4
ReaRAG	33.8	47.2	31.3	<u>47.6</u>	<u>26.1</u>	<u>39.1</u>	30.4	<u>44.6</u>
SynerGate-RAG	**45.0**	**58.5**	**63.8**	**68.7**	**28.8**	**42.5**	**45.9**	**56.6**

Among the systems in Table 1, SynerGate-RAG achieves the highest mean EM and F1 on all three benchmarks, with macro scores of 45.9 $\pm$ 0.3 and 56.6 $\pm$ 1.2, respectively. Its largest advantage occurs on 2WikiMultiHopQA, with 63.8 $\pm$ 0.9 EM and 68.7 $\pm$ 1.2 F1. These standard deviations summarize three evaluation runs; training-objective comparisons are reported in Table 2.

2WikiMultiHopQA combines Wikipedia and Wikidata with explicit reasoning paths (Ho et al., 2020), making it well suited to relation-based supervision. Its larger gain is consistent with graph feedback being more useful where intermediate deductions have explicit relational support. Under incomplete graph support, the text channel remains available. The ablation and deletion analyses below examine this interpretation; dataset provenance alone does not establish the quality of the supplied graphs or the cause of the gain.

4.3 CONTRIBUTIONS OF REWARD LEARNING AND ALLOCATION

Table 2: Matched training ablations. EM and F1 are reported in percent.

Training variant	HotpotQA		2Wiki		MuSiQue		Macro	
	EM	F1	EM	F1	EM	F1	EM	F1
Answer only	34.1	44.8	40.5	46.2	20.8	29.5	31.8	40.2
Text process only	38.2	51.1	45.3	52.0	25.1	36.0	36.2	46.4
Graph process only	35.4	47.6	58.2	63.8	20.6	30.1	38.1	47.2
Fixed graph–text mixture	38.6	52.0	56.5	62.1	24.2	35.5	39.8	49.9
Direct graph rules	37.2	50.1	57.9	63.2	22.8	33.7	39.3	49.0
Hard-label PRM	39.5	53.4	57.1	62.9	24.7	36.1	40.4	50.8
Hard-verdict gate	39.1	52.8	<u>59.4</u>	<u>64.8</u>	24.3	35.8	40.9	51.1
Terminal-only placement	<u>41.5</u>	<u>55.4</u>	58.2	63.9	<u>26.5</u>	<u>38.6</u>	<u>42.1</u>	<u>52.6</u>
SynerGate-RAG	**45.0**	**58.5**	**63.8**	**68.7**	**28.8**	**42.5**	**45.9**	**56.6**

Each variant changes the named factor from the full objective; all retain outcome reward and share initialization, evidence, and rollout budget. Fixed and graph-only allocation retain the availability mask. Hard labels keep neutral examples. SynerGate-RAG repeats the three-run mean from Table 1; other rows retain their reported ablation scores (Appendix C).

In Table 2, SynerGate-RAG exceeds answer-only training by 14.1 macro EM and 16.4 macro F1 points. The corresponding differences are 6.1/6.7 points relative to the training-mean fixed mixture and 6.6/7.6 points relative to direct graph rules. These comparisons favor the full objective in the reported macro scores. Hard-label training also scores lower while retaining the same samples, including neutral examples.

The full model outperforms both single-channel variants. Its F1 advantage over graph-only supervision is larger on MuSiQue (12.4 points) than on 2Wiki (4.9 points), consistent with a complementary role for textual feedback. Step-end placement also outperforms terminal-only placement despite an unchanged undiscounted reward total, supporting the role of temporal credit assignment.

4.4 Process-level reward quality

Table 3: Process-evaluation results.

Evaluator	Step preference accuracy	First-error accuracy	Utility MAE
Direct graph rules	68.5	54.0	0.184
Hard-label graph PRM	77.1	66.8	0.151
Continuous graph PRM	83.2	74.5	0.112

Accuracy is in percent; lower MAE is better. Utility MAE is against independent evidence-support ratings, not the rule labels used for training. First-error accuracy includes an explicit all-correct decision. The evaluation uses a RAG process set rather than ProcessBench.

Table 3 compares direct rule scoring with learned process evaluation. The continuous PRM achieves higher step-preference accuracy and first-error accuracy than either direct rule scoring or hard-label training. The process set includes relation substitutions and reversed comparisons with unchanged citations, as well as paraphrases that retain the same meaning. Final answers are hidden from the evaluator. These probes follow the distinction between answer selection and process-error identification emphasized in PRM evaluation (Zhang et al., 2025b; Zheng et al., 2025).

4.5 Independent utility of the gate

Table 4: Utility of process-only allocation on a common frozen candidate bank.

Allocation	Correct–wrong ranking	Top-1 answer EM
Fixed mean gate	79.8	65.2
Direct verdict strength	80.6	66.0
Shuffled learned gate	77.9	64.6
Learned gate	85.1	69.4

Ranking uses mixed-outcome question families, with ties assigned a score of 0.5; Top-1 EM includes all questions. The candidate generator and all scoring models are identical across rows. Direct verdict strength uses the rule target $|z_t|$, not a graph/text score ratio.

Relative to the fixed mean gate, learned allocation improves correct–wrong ranking by 5.3 points and Top-1 EM by 4.2 points. Shuffling within dataset and availability strata preserves the weight distribution but breaks its association with individual steps. The decline after shuffling supports the importance of this association, complementing the reader-training comparison in Table 2.

4.6 Robustness to incomplete graphs

Figure 3(a) shows less degradation under adaptive allocation than graph-only supervision. At complete graph removal, adaptive allocation, the masked fixed mixture, and text-only training share the same objective and result under shared initialization and randomness; graph-only training retains only outcome supervision. This training-time test therefore cannot attribute the graph-free endpoint to the gate.

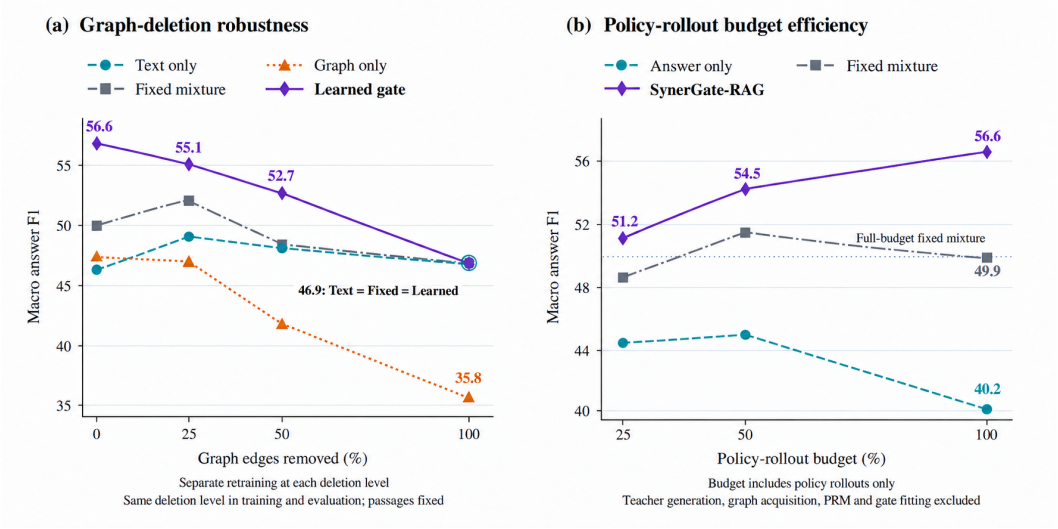

Figure 3: Graph-deletion robustness and policy-rollout efficiency. (a) Macro answer F1 under matched training/evaluation graph deletion; passages remain fixed. Each deletion level is retrained separately with shared initialization and masks across methods. (b) Macro answer F1 at matched policy-rollout budgets, excluding offline supervision preparation. The horizontal line denotes the full-budget fixed mixture.

4.7 Data efficiency and qualitative behavior

At 25% of the rollout budget, SynerGate-RAG reaches 51.2 F1, exceeding the full-budget fixed mixture (49.9; Figure 3(b)). The fixed-mixture curve is non-monotonic, so this comparison concerns its full-budget result rather than its best budget. Offline supervision and PRM/gate fitting are excluded; the curve does not establish lower total training cost. A schematic failure case further illustrates the method: a graph confirms two dates, yet the reader reverses their order while citing both correctly. Citation precision remains high despite the incorrect comparison; a process evaluator that uses the dates and the comparison can penalize the step. If one date is missing, the comparison cannot yield a contradiction witness; graph feedback is masked only if no other admissible source slot remains. This example is constructed for explanation and is not an observed model output.

5 Conclusions

SynerGate-RAG turns incomplete graph evidence into selective training feedback by learning signed utility and allocating graph–text rewards at each reasoning step. Across three benchmarks, it achieves the highest mean answer EM and F1 among the systems in Table 1, with the largest gains on 2WikiMultiHopQA. Training-objective comparisons and process evaluations support complementary roles for reward learning and allocation. Inference requires evidence preparation and the trained reader.

Supervision remains bounded by evidence coverage, entity linking, and weak-label quality. Verdict strength is an allocation proxy; additive step rewards remain sensitive to trajectory length and teacher–policy distribution shift. Further work can extend verifiable relations and operations and develop allocation targets that directly compare graph and text utility.

AI USE STATEMENT

In this work, we used generative AI tools for writing assistance, retrieval and discovery, and research ideation and execution. Writing assistance included drafting and polishing the manuscript, as well as checking wording, mathematical notation, citations, and formatting. Retrieval and discovery involved searching for relevant literature and identifying related work. Research ideation and execution included refining the methodological framework, mathematical formulations, and experimental design; modifying and reviewing research code; assisting with data-processing workflows; and interpreting diagnostic outputs. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Graph-grounded process supervision inherits errors, omissions, and representational biases from its evidence sources. Incorrect entity links or incomplete relations can produce misleading supervision, and availability gating cannot eliminate these risks. Answer accuracy and citation agreement should therefore be interpreted alongside the quality of the underlying evidence. The biographical examples in Appendix D are used to illustrate evidence use; they do not justify extending the method to sensitive personal-data inference. Any release of derived evidence, model outputs, or software must respect the applicable source licenses and privacy constraints.

REPRODUCIBILITY STATEMENT

Appendix A specifies the data partitions, supervision scales, model identities, retrieval configuration, and optimization settings. Appendix B details source matching, availability masking, the offline annotation prompt and contradiction checks, label construction, gate features, loss weights, and token-level optimization. Appendix C defines the ablation controls and graph-deletion conditions, while Appendix E specifies process-evaluation metrics, candidate scoring, and failure-handling rules. Appendix G provides the per-run results underlying Table 1 and the formulas used to aggregate them.

REFERENCES

AI@Meta. Llama 3 Model Card. Official model card, 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/25f7be9694d7b32d5cc670927b8091e1-Abstract-Conference.html.

Jie Cheng, Gang Xiong, Ruixi Qiao, Lijun Li, Chao Guo, Junle Wang, Yisheng Lv, and Fei-Yue Wang. Stop Summation: Min-Form Credit Assignment Is All Process Reward Model Needs for Reasoning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 131646–131671, 2025. doi: 10.52202/085713-4382. URL https://papers.neurips.cc/paper_files/paper/2025/hash/be91eb86eb74efc055cff83e953f86ce-Abstract-Conference.html.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. arXiv preprint arXiv:2404.16130v1, 2024. URL https://arxiv.org/abs/2404.16130v1.

Leo Gao, John Schulman, and Jacob Hilton. Scaling Laws for Reward Model Overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10835–10866. PMLR, 2023. URL https://proceedings.mlr.press/v202/gao23h.html.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022. URL https://iclr.cc/virtual/2022/poster/6319.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active Retrieval Augmented Generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7969–7992. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.495. URL https://aclanthology.org/2023.emnlp-main.495/.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 59532–59569, 2024. doi: 10.52202/079017-1902. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6ddc001d07ca4f319af96a3024f6dbd1-Abstract-Conference.html.

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

Zhicheng Lee, Shulin Cao, Jinxin Liu, Jiajie Zhang, Weichuan Liu, Xiaoyin Che, Lei Hou, and Juanzi Li. ReaRAG: Knowledge-guided Reasoning Enhances Factuality of Large Reasoning Models with Iterative Retrieval Augmented Generation. arXiv preprint arXiv:2503.21729, 2025. URL https://arxiv.org/abs/2503.21729.

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. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. In *International Conference on Learning Representations*, pp. 39578–39601, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. Multi-Hop Knowledge Graph Reasoning with Reward Shaping. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3243–3253. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1362. URL https://aclanthology.org/D18-1362/.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning. In *International Conference on Learning Representations*, pp. 14400–14423, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/3e2aeb66481dd63a32421bf032b70384-Abstract-Conference.html.

Miao Peng, Nuo Chen, Zongrui Suo, and Jia Li. Rewarding Graph Reasoning Process makes LLMs more Generalized Reasoners. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, volume 2, pp. 2257–2268. Association for Computing Machinery, 2025. doi: 10.1145/3711896.3737109. URL https://doi.org/10.1145/3711896.3737109.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1506.02438.

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

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-Searcher: Incentivizing the Search Capability in LLMs via Reinforcement Learning. arXiv preprint arXiv:2503.05592, 2025. URL https://arxiv.org/abs/2503.05592.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Is Multihop QA in DiRe Condition? Measuring and Reducing Disconnected Reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8846–8863. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.712. URL https://aclanthology.org/2020.emnlp-main.712/.

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. URL https://aclanthology.org/2022.tacl-1.31/.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 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)*, pp. 10014–10037. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.557. URL https://aclanthology.org/2023.acl-long.557/.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable Preferences via Multi-Objective Reward Modeling and Mixture-of-Experts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10582–10592. Association for Computational Linguistics, 2024a. doi: 10.18653/v1/2024.findings-emnlp.620. URL https://aclanthology.org/2024.findings-emnlp.620/.

Liang Wang, Haonan Chen, Nan Yang, Xiaolong Huang, Zhicheng Dou, and Furu Wei. Chain-of-Retrieval Augmented Generation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 59888–59915, 2025. doi: 10.52202/085713-2003. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/566aad4fac1acf17fd1ae8c3aef75326-Abstract-Conference.html.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-Shepherd: Verify and Reinforce LLMs Step-by-step without Human Annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439. Association for Computational Linguistics, 2024b. doi: 10.18653/v1/2024.acl-long.510. URL https://aclanthology.org/2024.acl-long.510/.

Xinyi Wang, Yiping Song, Zhiliang Tian, Bo Liu, Tingjin Luo, and Minlie Huang. DPRM: A Dual Implicit Process Reward Model in Multi-Hop Question Answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(40):33683–33691, 2026. doi: 10.1609/aaai.v40i40.40658. URL https://ojs.aaai.org/index.php/AAAI/article/view/40658.

Yilin Wen, Zifeng Wang, and Jimeng Sun. MindMap: Knowledge Graph Prompting Sparks Graph of Thoughts in Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10370–10388. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.558. URL https://aclanthology.org/2024.acl-long.558/.

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

Xiaopeng Ye, Chen Xu, Chaoliang Zhang, Zhaocheng Du, Jun Xu, Gang Wang, and Zhenhua Dong. Q-PRM: Adaptive Query Rewriting for Retrieval-Augmented Generation via Step-level Process Supervision. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 15113–15128. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.817. URL https://aclanthology.org/2025.findings-emnlp.817/.

Chuanyue Yu, Kuo Zhao, Yuhan Li, Heng Chang, Mingjian Feng, Xiangzhe Jiang, Yufei Sun, Jia Li, Yuzhi Zhang, Qingyun Sun, Jianxin Li, and Ziwei Zhang. GraphRAG-R1: Graph Retrieval-Augmented Generation with Process-Constrained Reinforcement Learning. In *Proceedings of the ACM Web Conference 2026*, pp. 1398–1409. ACM, 2026. doi: 10.1145/3774904.3792589. URL https://doi.org/10.1145/3774904.3792589.

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*, volume 38, pp. 58701–58729, 2025a. doi: 10.52202/085713-1965. URL https://papers.neurips.cc/paper_files/paper/2025/hash/54e1381d0c0598127b90af4c940fd3d9-Abstract-Conference.html.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The Lessons of Developing Process Reward Models in Mathematical Reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10495–10516. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.findings-acl.547. URL https://aclanthology.org/2025.findings-acl.547/.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. ProcessBench: Identifying Process Errors in Mathematical Reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1009–1024. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.50. URL https://aclanthology.org/2025.acl-long.50/.

Xiangrong Zhu, Yuexiang Xie, Yi Liu, Yaliang Li, and Wei Hu. Knowledge Graph-Guided Retrieval Augmented Generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8912–8924. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.449. URL https://aclanthology.org/2025.naacl-long.449/.

A DATA, MODELS, AND SETTINGS

The reader and graph PRM use Meta-Llama-3-8B-Instruct (AI@Meta, 2024), with separate adaptations for generation and process evaluation.

The offline supervision used a teacher pool of 10,000 questions and one requested trajectory per question, giving 10,000 requested teacher trajectories before parsing, annotation, and filtering. This preparation budget is distinct from the policy-training budget: 6,000 questions balanced across the three datasets, ten passages per question, and four on-policy trajectories per question, for 24,000 policy trajectories. A separate 2,000-example pool supplies supervised replay. Requested teacher trajectories do not determine the number of accepted SFT demonstrations or retained PRM and gate steps.

Teacher generation, PRM fitting, gate fitting, and process evaluation use explicit question-family partitions. Reader initialization uses intact, accepted trajectories: deleting isolated erroneous steps from a dependent chain is avoided. PRM examples include source-backed negative and abstained steps, with all variants of a parent question kept in one partition.

Table 5: Offline supervision scale and data partitions.

Stage	Families	Trajectories	Steps	Negative	Zero	Positive
Teacher requests	10,000	10,000	39,581	—	—	—
Accepted SFT	4,028	4,028	16,187	—	—	—
PRM training	5,500	5,500	21,786	2,307	6,629	12,850
PRM development	700	700	2,789	287	868	1,634
Gate training	2,500	2,500	9,937	973	3,094	5,870
Gate development	300	300	1,204	131	349	724

The table reports the recorded offline supervision counts. The teacher row reports requested trajectories and the source-step slots obtained from the teacher pool; the request count does not itself specify the number of successful generations. Steps are source-step slots before perturbation augmentation. Negative, zero, and positive refer to the original graph target z_t; the gate is trained on $|z_t|$. SFT is an overlapping subset, so rows must not be summed.

The 4,028 accepted SFT trajectories correspond to a 40.28% acceptance rate relative to 10,000 teacher requests. The four PRM and gate partitions are family-disjoint and contain 9,000 families and 35,716 source-step slots in total. The remaining 1,000 families and 3,865 slots form an unallocated reserve and are not counted as annotation failures.

The PRM and gate training union contains 8,000 families. All accepted SFT examples are placed within this union; 2,000 of them supply replay, and the remaining 6,000 families supply policy questions, with no policy–replay family overlap. Component-development partitions and the reserve are excluded from these training uses. Reader-development and reporting questions lie outside the teacher pool.

The evaluation design separates a 900-question reporting population, with 300 questions per dataset, from a 150-question development population. EM and F1 use accepted answer aliases under a shared versioned scorer. Checkpoint selection is confined to development data.

Table 1 uses evaluation seeds 42, 43, and 44, with 300 questions per dataset in each run. These are benchmark-evaluation seeds, distinct from reader-training seeds. Decoding uses temperature zero with sampling disabled; the per-call generation cap is 1,024 tokens for R1-Searcher and 512 for other methods. Table 1 reports the arithmetic mean over the three runs; sample standard deviations are summarized in the result discussion. Macro scores first average the three datasets within each run; their standard deviation is then computed across the three run-level macro scores, not by averaging dataset-level deviations. Appendix G lists the individual results.

Table 6: Optimization settings.

Quantity	Value
Reader / graph PRM backbone	Llama-3-8B-Instruct
Parameter-efficient adaptation	LoRA; rank 16, scaling 32, dropout 0.05
PRM / gate objectives	Weighted Huber / soft-target BCE
PRM Huber delta / epochs	0.2 / 3
PRM learning rate / effective batch	5×10^{-5} / 16
Gate epochs / learning rate	100 / 1×10^{-3}
Gate temperature	Learned; lower bound 0.1
Policy trajectories / effective batch	24000 / 4
Policy mini-batch / PPO epochs	1 / 1
Policy learning rate / clip	1×10^{-6} / 0.2
Discount / GAE lambda	0.95 / 0.95
Value-loss weight / initial KL coefficient	0.5 / 0.15
Answer / step / text scales	4.0 / 1.5 / 0.3
Replay coefficient / interval	0.10 / every 10 policy batches
Maximum gradient norm	1.0; policy and replay updates
Reader evaluation decoder	Greedy; up to 512 new tokens
Reader-training seeds	42, 43, 44
Evaluation seeds (Table 1)	42, 43, 44

Parameter-efficient adaptation follows LoRA (Hu et al., 2022). The answer, step, and text scales correspond to $\lambda_A = 4.0$, $\lambda_{\text{step}} = 1.5$, and $c_D = 0.3$, respectively. Thus the text term in Equation (5) has coefficient $0.45(1 - \alpha_t)$, becoming 0.45 under textual fallback. The text-score transformation, $r_t^D = \tanh\left(\frac{2.5-\ell_t}{1.5}\right)$, remains fixed across comparisons.

A.1 Evidence sources and supervision models

Table 7: Evidence and supervision components.

Component	Configuration
Passage corpus	Wiki18
Dense retriever	intfloat/e5-base-v2
Sparse retriever	BM25 with the bm25s backend
Fusion / reranker	RRF (rank constant 60); bge-reranker-v2-m3
Knowledge graph	Wikidata; SPARQL; up to two relation hops
Teacher / offline annotator	deepseek-v4-flash; temperature 0
Text scorer	THU-KEG/ReaRAG-9B; frozen

The reader and graph PRM backbones and optimization settings appear in the optimization table.

Passage retrieval uses E5-base-v2 and BM25 to obtain 100 candidates each. Reciprocal-rank fusion with a rank constant of 60 retains 50 passages, which BGE-reranker-v2-m3 reranks to select ten. The passage context has a budget of 3,860 tokens. Evidence selection uses the question and retrieved content, without reference answers, gold supporting facts, or annotated reasoning paths.

Wikidata evidence is acquired during evidence preparation. Mentions in the question and retrieved passages are linked to Wikidata entities, and SPARQL queries collect a local subgraph within two relation hops. Question-conditioned filtering retains at most 12 source-backed triples under a shared budget for the teacher, reader, and process scorers. Statement-level records preserve canonical entity and property identifiers, typed values, and relevant qualifiers. Missing relations do not constitute contradictions, and passage-derived navigation edges receive no graph-reward credit.

The evidence collection is frozen per question before trajectory generation or evaluation. Its cache retains the exact source statements, statement and source-revision identifiers, retrieval timestamps, and content hashes; model inputs expose only the retained facts and their relevant qualifiers. Controlled reward variants reuse the same cached evidence, and graph-deletion conditions derive their views from these fixed inputs. This collection is a frozen set of retrieved Wikidata statements, with a temporal scope separate from Wiki18. SynerGate-RAG performs no graph retrieval or teacher/annotator API calls during PPO.

A.2 Operational specification for evidence construction

Evidence construction follows a fixed interface and deterministic selection rules. The relation registry defines the scope of admissible graph evidence; facts outside this scope receive no graph credit.

Entity candidates come from a frozen English Wikidata label/alias index. Apply Unicode NFKC, case folding, and whitespace collapse, then select longest whole-alias matches from left to right; a whole-alias boundary has no adjacent Unicode letter or digit. A unique candidate QID is accepted. An ambiguous alias is resolved only when exactly one of its candidates is identified by an exact English-Wikipedia sitelink for a retrieved passage title; otherwise it remains unresolved. Duplicate QIDs retain their first occurrence. Question anchors precede passage-only anchors, which follow passage rank and mention position. After graph selection, canonical labels and aliases of retained graph nodes complete the question-specific resolution table. This table is frozen and shared by citation matching, source availability, and gate features. Generated text supplies lookup keys only; it cannot extend this table or establish a fact.

Table 8: Fixed relation registry in the operational specification.

Property	English aliases	Object type
P57	director; directed by	Entity
P50	author; written by	Entity
P22	father	Entity
P25	mother	Entity
P569	date of birth; birth date; born	Time
P570	date of death; death date; died	Time
P19	place of birth; birthplace	Entity
P20	place of death; deathplace	Entity
P17	country	Entity
P27	country of citizenship; citizenship; nationality	Entity
P131	located in the administrative territorial entity; administrative territory	Entity
P159	headquarters location; headquarters	Entity
P1082	population; inhabitants	Quantity

These aliases define the English-language task interface. The registry supplies retrieval properties and the relation keys used by the availability mask. No inverse navigation operator or additional property is implicit.

Relation aliases use the same longest-match rule. For comparison-slot extraction, the fixed operators are $<$ (before, earlier than), $\leq$ (no later than, at most), $>$ (after, later than), $\geq$ (no earlier than, at least), $=$ (equal to, same as), and $\neq$ (not equal to, different from), together with their literal mathematical symbols. Time slots require two resolved entity operands and values of the same time property and calendar model; quantity slots require the same property and unit. All applicable qualifiers are retained. Missing or ambiguous operands leave that slot incomplete; they do not suppress another complete slot. The witness checks in Appendix B determine whether a source-ready comparison is semantically decidable. Unregistered paraphrases and operations remain outside the graph-verification scope.

SPARQL retrieves non-deprecated statement records using only the registered properties, following their original subject-to-object direction for at most two semantic relation hops; statement nodes do not add a hop, and literal objects terminate expansion. After the source-identity and uniqueness checks below, enumerate simple one- and two-hop paths from each anchor. For a path P, let $k_q(P)$ count its distinct properties whose aliases occur in the question. Sort paths by $(-k_q(P), |P|, \mathrm{IDs}(P))$, with statement-ID sequences compared lexicographically. Process question anchors first; begin passage-only anchors when no question-anchor path can add a fact. Within each group, visit anchors in round-robin order and take the next path that adds a fact while fitting the 12-fact cap. Consume each candidate path at most once. Add accepted paths in full and count shared facts once; stop at the cap or after both groups are exhausted. This deterministic ordering uses neither answer labels nor supporting-fact annotations; it favors reproducibility and complete short paths rather than guaranteeing semantic relevance.

Before path selection, canonicalize (h, r, o) using QIDs, PIDs, and typed literals. Records with the same triple and the same complete normalized qualifier set are merged into one fact while retaining every source statement/revision ID; the lexicographically smallest statement ID represents the merged fact for path ordering. If an otherwise identical triple has distinct qualifier sets, exclude that ambiguous group from the derived candidate graph and retain its raw records separately. Each final graph line displays a local fact ID, its three fields, and its qualifiers. A reverse index maps a copied triple to its unique retained source record; explicit or resolved canonical identities must agree, and a non-unique or missing match receives $V_{ti} = 0$. This preserves the three-field citation format while making the inherited qualifiers and provenance unambiguous. Figure 2 abbreviates this attached metadata.

The scoring views are $D_t = D$ and $G_t = G$ for every step: the same rendered passage block and final retained graph accompany each prefix and current step. The passage-token budget is counted with the reader tokenizer, including passage titles and separators. A scorer may access only the statement IDs retained in its current graph view and the corresponding metadata. Raw candidate records, excluded qualifiers, pre-deletion neighborhoods, and cached facts outside that view cannot enter its prompt, features, source matching, or witnesses. Raw entity JSON and provenance documents remain audit-only. Graph deletion rebuilds the fact-ID allowlist, reverse index, density, and availability from the remaining facts; it does not refill the graph. The alias resolver can identify mentions, but alias entries alone never supply a relation. An empty graph therefore retains the exact textual fallback of Equation (5).

B Label semantics and token-level optimization

B.1 Source matching and the availability mask

Let R_t be the current reasoning and conclusion with the citation field removed, and $C_t = (c_{t1}, \ldots, c_{tn_t})$ its declared citation items. An admissible fact is a record $(h, r, o, \eta, \mathrm{id})$ containing canonical entity and relation identities, typed values, qualifiers η, and a resolvable source revision. Matching normalizes Unicode, whitespace, and typed literals, then resolves labels through the frozen alias map. Entity and relation identities, direction, literal units, and applicable qualifiers must agree. Ambiguous aliases and unmatched or malformed items receive $V_{ti} = 0$; lexical similarity alone cannot establish a match. All declared items, including unmatched items, remain in the denominator.

The availability check uses question, prefix, and current-step mentions, excluding the citation field. Unicode-normalized, case-folded longest whole-alias matches identify candidate entities and relations; tied identities are unresolved. The frozen relation/operator registry names aliases, typed comparison slots (such as birth dates), and the operands and qualifiers each slot requires. Let E_t and P_t be the resulting entity and relation sets. A source fact enters $\mathcal{A}_t$ only if it is incident to E_t, has relation in P_t, passes source-identity checks, and has the typed values and qualifiers needed for that operation. A comparison slot requires both operands; incomplete slots are excluded individually. Mentions in earlier generated text are lookup keys, not accepted facts. Set $m_t = \mathbf{1}[\mathcal{A}_t \neq \varnothing]$. The mask is zero only when no complete admissible slot remains; record source, scope, operand, and provenance failures separately. One incomplete slot does not suppress a different complete slot. Neither a fixed edge-count threshold nor the presence of citations determines m_t.

This is a conservative check of source readiness, not an entailment decision. An available source can still leave a semantic judgment unresolved. Offline annotation and PPO use the same frozen scope extractor and source contract. In full SynerGate-RAG, PPO does not rerun the teacher or compute v_t and b_t; the learned PRM supplies the signed graph score. The hard mask only controls whether that score is admissible. Purely implicit deductions without an admissible entity–relation slot receive textual feedback through $m_t = 0$; the graph branch does not verify unrestricted commonsense inference.

B.2 Offline semantic annotation and witness validation

Teacher trajectory generation and offline semantic annotation use deepseek-v4-flash with temperature zero. The annotator uses the same fixed model as the teacher and is restricted to the supplied evidence. It aligns free-text claims with candidate premises and proposes structured checks; witness acceptance requires validated references to supplied source facts. The prompt below is applied once per step. Runtime inputs are serialized as JSON in source order, with no reference answer. The model identifier, prompt hash, alias-map and operator-registry revisions, and decoding settings are fixed for the label version. Decisions are categorical, with explicit uncertainty and no entailment-probability threshold.

Offline annotation prompt (in English).

```
Role: offline graph-utility annotator. Use only supplied evidence.
Inputs: question; current reasoning and conclusion (citations
↪ removed);
declared citations; candidate source fact IDs with canonical
↪ fields and qualifiers;
previous steps as context only, never as factual authority.
Do not use a reference answer, outside knowledge, or future steps.

1. For every citation, return RELEVANT, IRRELEVANT, or UNKNOWN.
   RELEVANT means its asserted relation supplies a premise or
   ↪ operand
   used by the current claim. Mere entity overlap is insufficient.
   Judge its role separately from whether the citation matches a
   ↪ fact.
   Quote the exact current-step span and name the role:
   ↪ premise/operand.
2. Extract explicit current claims; retain direction, polarity,
↪ values,
   units, and temporal qualifiers. Do not complete an unstated
   ↪ claim.
3. Propose contradictions only for NEGATED_FACT or
↪ REVERSED_COMPARISON.
   Give the exact claim span, canonical operands, and source fact
   ↪ IDs.
   A missing edge or a different tail alone is never a
   ↪ contradiction.
   For comparisons, give the operator and both source-backed
   ↪ values.
4. Return UNKNOWN when wording, identity, scope, or evidence is
↪ unclear.
   Do not assign a final scalar reward or an answer-correctness
   ↪ score.
5. Return one JSON object; no prose outside it. Use this schema:
   relevance: [{citation_index, verdict, claim_span, role}]
   claims: [{claim_span, subject, relation, object, polarity,
   ↪ qualifiers}]
   witnesses: [{claim_span, rule, fact_ids, operator, operands}]
   unknown_reasons: [reason]
```

The field listing above denotes the object structure; serialized keys and string values use double quotes. An abstaining annotation with no declared citations or extractable claims can take the following form (format example):

```
{
  "relevance": [],
  "claims": [],
  "witnesses": [],
  "unknown_reasons": ["Current claim is ambiguous."]
}
```

For relevance, accept $L_{ti} = 1$ only for RELEVANT with a valid citation index, an exact nonempty span in R_t, and a premise or operand role. IRRELEVANT and UNKNOWN give $L_{ti} = 0$. Each citation must receive exactly one decision; missing decisions, invalid fields, duplicate indices, or out-of-range indices invalidate the response. Relevance assesses the claimed role of a citation even when that citation is unverified. The two marginal ratios are therefore

$$p_t = \frac{\sum_{i=1}^{n_t} V_{ti}}{\max(1, n_t)}, \qquad v_t = \frac{\sum_{i=1}^{n_t} L_{ti}}{\max(1, n_t)}.$$

Under Equation (1), $p_t v_t$ need not equal the fraction of citations that are both verified and relevant. For two citations, one verified but irrelevant and the other relevant but unverified, $p_t = v_t = 1/2$ gives $z_t = 1/4$ although no citation passes both checks. Per-citation joint decisions are retained for audit; the target remains a weak utility proxy. Thus, the target measures composite graph utility rather than per-citation joint support. In contrast to Monte Carlo–derived continuation-outcome labels (Wang et al., 2024b; Zhang et al., 2025b), these labels describe supplied-evidence use.

A contradiction witness must quote a current-step claim and reference fact IDs in the admissible candidate scope. Each operand is resolved directly to source facts; generated earlier conclusions are never accepted as independent evidence. The checker accepts only two operations. NEGATED_FACT requires opposite polarity for the same canonical subject, relation, object, and qualifiers. REVERSED_COMPARISON evaluates an explicit order or equality operator on two source-backed numeric values with the same unit or dates of compatible precision, and accepts a witness only when the stated comparison is false. Possible date intervals must make the comparison decidable; overlapping intervals return UNKNOWN. Different tails do not imply conflict, even for a relation that is often single-valued. Identity ambiguity, changed qualifiers, unsupported operations, missing operands, and explicit expressions of uncertainty do not constitute contradictions.

Let $\mathcal{W}_t$ contain accepted witnesses. Define $b_t = m_t \mathbf{1}[\mathcal{W}_t \neq \varnothing]$, which enforces $b_t \leq m_t$. Witness search examines R_t even when C_t is empty. The preceding text supplies context for resolving mentions, but previous conclusions and their positive utility labels do not constitute evidence for new facts. LLM claim alignment may be incorrect even for a well-formed witness, so these checks constrain weak supervision rather than certify arbitrary natural-language entailment. Unrecognized paraphrases and operations outside the two-rule inventory are abstained from.

B.3 Label assembly, failures, and records

- Parse the step, compute deterministic citation matches and m_t, and retain all parser diagnostics. If $m_t = 0$, set $z_t = 0$ and bypass semantic annotation.
- For $m_t = 1$, run the offline annotation instruction once and validate its JSON and source references. On timeout, truncation, invalid JSON, duplicate decisions, or schema failure, retain the failed record but do not assign a training target. Exclude it from PRM and gate fitting (zero PRM loss weight), and retain m_t as the independent source-readiness result.
- For a valid annotation, compute relevance decisions and accepted contradiction witnesses. Apply Equation (1) in order: accepted contradiction gives -1; otherwise use $p_t v_t$ when the source is available; an unavailable source gives zero. Do not consult the final answer.
- Store question family, partition, step and prefix, source revisions, alias-map, operator-registry, and prompt hashes, raw annotation response, matched facts, per-citation decisions,

witness validation results, reason codes, and $(p_t, v_t, b_t, m_t, z_t)$. Keep targetless annotation failures separate from genuine zero-utility labels.

PRM and gate sample counts are computed after parsing and family assignment, not from the number of requested teacher completions. Raw completion counts and retained-step counts describe different stages of supervision construction. Any class-balancing or confidence weights are fixed on the training partition; final test answers do not enter weight selection. The annotator, source snapshot, alias map, operator registry, prompt, witness rules, and failure policy are fixed within each supervision dataset.

B.4 Operational specification for features and loss weights

This operational specification completes the numerical definitions in Equations (2)–(4), without changing the utility target or graph–text reward. Let $\mathcal{V}(G)$ contain the canonical entity and typed-literal nodes incident to retained facts, let $n = |\mathcal{V}(G)|$, and let $\mathcal{E}_*(G) = \{(h, o) : (h, r, o) \in G,\ h \neq o\}$. Parallel properties and repeated source statements count once in the directed node-pair density:

$$d(G) = \begin{cases} |\mathcal{E}_*(G)|/[n(n-1)], & n \geq 2, \\ 0, & n < 2. \end{cases}$$

Let M_t be the distinct normalized entity-alias strings detected in R_t, the current reasoning and conclusion with citations removed. Let $J_t \subseteq M_t$ contain those resolved to one QID by the frozen question-specific resolver. Define $c_t = |J_t|/\max(1, |M_t|)$. Unresolved ties contribute zero; repeated occurrences do not increase their weight. This is a link-resolvability proxy, not a calibrated probability of semantic correctness. For nonempty parsed step-token positions S_t, excluding padding and the final-answer field, the frozen reference reader gives

$$h_t = -\frac{1}{|S_t|} \sum_{j \in S_t} \log \pi_{\mathrm{ref}}(y_j \mid q, D, G, y_{<j}).$$

The step spans include their generated schema and citation tokens; the conditioning prefix contains no future text. Density can remain constant across steps of one question, whereas c_t and h_t are recomputed for each realized step. For raw features $u_t = (d(G), c_t, h_t)$, let $\mathcal{I}_g$ be the valid gate-training examples and $N_g = |\mathcal{I}_g| > 0$. Compute and freeze their population statistics:

$$\mu_k = \frac{1}{N_g} \sum_{t \in \mathcal{I}_g} u_{tk}, \qquad \sigma_k = \sqrt{\frac{1}{N_g} \sum_{t \in \mathcal{I}_g} (u_{tk} - \mu_k)^2}.$$

Use $\widetilde{u}_{tk} = (u_{tk} - \mu_k)/\sigma_k$ for $\sigma_k > 0$, or zero for a constant feature. Gate fitting retains the unweighted soft-target BCE in Equation (4).

If fitting uses coefficients $\widetilde{W}, \widetilde{b}$ on standardized features, the effective coefficients below absorb the fixed affine transform. Equation (3) therefore keeps its raw-feature form, including the learned temperature and availability mask:

$$W_k = \begin{cases} \widetilde{W}_k/\sigma_k, & \sigma_k > 0, \\ 0, & \sigma_k = 0, \end{cases} \qquad b = \widetilde{b} - \sum_{k:\sigma_k > 0} \widetilde{W}_k \mu_k/\sigma_k.$$

For PRM balancing, partition valid training targets into $\mathcal{C}_- = \{t : z_t = -1\}$, $\mathcal{C}_0 = \{t : z_t = 0\}$, and $\mathcal{C}_+ = \{t : z_t > 0\}$, and let $c(t)$ denote the class of step t. With class sizes N_c, total size $N > 0$, and K nonempty classes, assign the following fixed training-set weights in Equation (2):

$$w_t = \frac{N}{K\, N_{c(t)}} \quad (t \text{ has a valid target}), \qquad w_t = 0 \quad (t \text{ is an annotation failure}).$$

Counts are fixed after training-partition filtering and are not recomputed by minibatch or from development/test labels. Targetless annotation failures are excluded from the loss and class counts. Source-unavailable zero targets remain in $\mathcal{C}_0$, and positive targets retain their original continuous values. Each nonempty class has equal total loss weight; this balances frequency rather than certifying label reliability and can amplify errors in a rare class.

B.5 Token-level policy optimization

The format predicate requires consecutive nonempty steps with reasoning and conclusion fields and a single final-answer field. Valid structured responses contain two to six steps; graph citations may be empty. The six-step limit belongs to $V(y)$ and does not silently truncate process scoring. All parseable steps retain their process terms, including those in an overlength or otherwise malformed response, while $V(y) = 0$ adds the terminal invalid penalty. Among task-reward terms, a wholly unparseable response receives only that penalty. The invalid penalty therefore does not guarantee that an invalid response has a negative total reward. These rules are shared across controlled reward variants and preserve Equation (6).

Let j_t be the last token of step t, and let L be the response length. The actual token reward combines step feedback, the terminal term r^{end}, and the reference-policy penalty:

$$\widehat{r}_j = \sum_t \mathbf{1}[j = j_t] r_t^{\text{proc}} + \mathbf{1}[j = L] r^{\text{end}} - \beta \log \frac{\pi_{\text{old}}(y_j \mid y_{<j}, q, D, G)}{\pi_{\text{ref}}(y_j \mid y_{<j}, q, D, G)}. \tag{7}$$

$$\delta_j = \widehat{r}_j + \gamma V_\nu(j+1) - V_\nu(j), \qquad \widehat{A}_j = \sum_{l \geq 0} (\gamma \lambda_{\text{GAE}})^l \delta_{j+l}. \tag{8}$$

The terminal value is zero, and the sum stops at the response end. PPO uses the clipped likelihood-ratio surrogate with these advantages and a separately trained value head (Schulman et al., 2017; 2016). Reward models do not share updated parameters with the policy. Supervised replay is a separate loss on the reference demonstrations; gradient clipping is applied on every update, including replay. Reward-component magnitudes, KL, advantage statistics, value loss, clipping frequency, valid-output rate, and step count are recorded together, since a rising proxy reward can accompany worse answers (Gao et al., 2023).

Temporal aggregation is a separate design choice. PURE studies reward exploitation under summed process feedback and proposes min-form credit assignment (Cheng et al., 2025). Our gate controls graph–text allocation within each step; it retains additive temporal rewards and does not by itself resolve this risk.

C Controlled analyses and reward sensitivity

For the training-step collection $\mathcal{T}_{\text{tr}}$, the fixed-mean control uses $\bar{g} = \sum_{t \in \mathcal{T}_{\text{tr}}} m_t g_\omega(u_t) / \sum_{t \in \mathcal{T}_{\text{tr}}} m_t$ and $\alpha_t^{\text{fixed}} = m_t \bar{g}$, whenever the denominator is positive. A source-unavailable step has zero masked graph allocation. This control replaces step-conditioned allocation by its training mean; it is not a development-tuned constant. The direct-verdict gate uses $m_t |z_t|$ from the annotator, and shuffled allocation permutes learned outputs within dataset and availability strata.

The hard-label PRM maps targets at thresholds -0.5 and 0.5 to negative, neutral, and positive classes, retains all samples, and converts predicted probabilities to a signed expected score. Hard gate supervision replaces $|z_t|$ with an indicator of a non-neutral verdict. The direct-rule ablation replaces only the PRM reward and retains the fitted gate. These definitions avoid conflating target discretization with data deletion.

Reward-side and joint graph deletion affect different graph views. In a reward-side study, passages and reader-visible graphs stay fixed while only the supervisor graph is degraded. In a joint study, both training and evaluation graph views are degraded and the reader is retrained. Figure 3(a) specifies the latter with shared initialization and identical deletion masks across methods; its empty-graph endpoint enforces the identity of adaptive and text-only objectives. Deletion rates refer to edges sampled from the supplied graph, not verified semantic-coverage percentages.

The text-only variant fixes $\alpha_t = 0$. The graph-only variant sets the text channel to zero and uses $\alpha_t = m_t$; it does not inherit textual fallback. The answer-only variant removes all process terms. Gate diagnostics use the mean process score across parsed steps on a common frozen reader bank with four candidates per question; invalid responses rank last, and ties use the original candidate index. Process-preference pairs share the same question and prefix. The first-error decision uses a threshold chosen only on the process-development split. No judge is given final answers when assessing individual steps.

Equation (5) yields a simple bounded-sensitivity statement. If both channel scores lie in $[-1, 1]$ and the scale coefficients are nonnegative, then for any two gate values on the same step,

$$|r^{\mathrm{proc}}(\alpha) - r^{\mathrm{proc}}(\alpha')| \le \lambda_{\mathrm{step}}(1 + c_D)|\alpha - \alpha'|. \tag{9}$$

This follows by factoring the difference as $\lambda_{\mathrm{step}}(\alpha - \alpha')(r^G - c_D r^D)$ and applying the triangle inequality. In addition, $m_t = 0$ implies exact textual fallback. This local upper bound on reward sensitivity and the fallback identity do not guarantee policy performance. A learned gate may be misspecified, and changing process rewards can change the optimal policy.

D Recorded question-answering cases

We present three responses generated by the reader trained with SynerGate-RAG, each with a correct final answer and explicit graph citations. All three questions are from 2WikiMultiHopQA and were selected retrospectively from the 60-question training-side diagnostic bank. Each response was generated using greedy decoding with the original ten retrieved passages and graph input. These examples illustrate graph-supported reasoning; their retrospective selection does not support conclusions about overall performance.

The inputs, generations, and scores in the source diagnostic bank passed independent verification. For readability, we condense the recorded reasoning and quote each triple field to make entity boundaries explicit. Facts, step order, cited relations, and final answers are preserved. All displayed triples occur in the reader-visible graph; the unedited responses and source hashes remain in the accompanying records.

D.1 Case D1. Comparing directors through their birth dates

Question: Which film has the director who was born first, Dracula 3D or Austerlitz (2016 Film)?

Evidence. The visible graph links Dracula 3D to Dario Argento and Austerlitz to Sergei Loznitsa, then supplies the two directors' birth dates. The four supplied relations support both entity identification and the temporal comparison.

Reasoning with graph evidence

Step 1. Identify the directors.
Dracula 3D was directed by Dario Argento; Austerlitz was directed by Sergei Loznitsa.
Graph evidence:
("Dracula 3D", "director", "Dario Argento")
("Austerlitz", "director", "Sergei Loznitsa")

Step 2. Compare birth dates.
Argento was born on 7 September 1940, before Loznitsa, who was born on 5 September 1964.
Graph evidence:
("Dario Argento", "date of birth", "7 September 1940")
("Sergei Loznitsa", "date of birth", "5 September 1964")

Step 3. Resolve the film.
The earlier-born director is Argento, whose film in the question is Dracula 3D.
Graph evidence:
("Dracula 3D", "director", "Dario Argento")

Answer: Dracula 3D (matches the reference). EM / F1: 100.0 / 100.0.

The first step identifies the directors, the second establishes the earlier birth date, and the third maps that director back to the film. Matching the citations alone would not determine whether the comparison is correct; the recorded conclusion also follows the order of the supplied dates.

D.2 Case D2. Following a film-to-director-to-birth-date chain

Question: When was the director of film Recep İvedik 2 born?

Evidence. The graph supplies two connected facts: Recep İvedik 2 was directed by Togan Gökbakar, whose date of birth is 29 May 1984. The director is the intermediate entity linking the film to the requested date.

Reasoning with graph evidence

> **Step 1. Identify the director.**
> Togan Gökbakar directed Recep İvedik 2.
> *Graph evidence:*
> ("Recep İvedik 2", "director", "Togan Gökbakar")
>
> **Step 2. Read the birth date.**
> Gökbakar was born on 29 May 1984, which answers the question.
> *Graph evidence:*
> ("Togan Gökbakar", "date of birth", "29 May 1984")

Answer: 29 May 1984 (matches the reference). EM / F1: 100.0 / 100.0.

The reader resolves the intermediate director before returning the birth date. Both steps cite their corresponding supplied relations, making the transition from the film entity to the answer explicit.

D.3 Case D3. Following a father relation to a death date

Question: What is the date of death of Peter Broadbridge, 3rd Baron Broadbridge's father?

Evidence. The graph identifies Eric Broadbridge, 2nd Baron Broadbridge as the father of Peter Broadbridge, 3rd Baron Broadbridge, and supplies Eric's date of death as 18 November 1972.

Reasoning with graph evidence

> **Step 1. Identify the father.**
> Peter Broadbridge's father is Eric Broadbridge, 2nd Baron Broadbridge.
> *Graph evidence:*
> ("Peter Broadbridge, 3rd Baron Broadbridge", "father", "Eric Broadbridge, 2nd Baron Broadbridge")
>
> **Step 2. Read the death date.**
> Eric Broadbridge died on 18 November 1972.
> *Graph evidence:*
> ("Eric Broadbridge, 2nd Baron Broadbridge", "date of death", "18 November 1972")
>
> **Step 3. Return the requested date.**
> The requested date of Peter Broadbridge's father's death is therefore 18 November 1972.
> *Graph evidence:*
> ("Eric Broadbridge, 2nd Baron Broadbridge", "date of death", "18 November 1972")

Answer: 18 November 1972 (matches the reference). EM / F1: 100.0 / 100.0.

The reader first resolves the father and then retrieves the date attached to that person. Its last step restates the resolved date rather than adding a new factual hop. This distinction separates the two-hop evidence chain from the three-step output format.

E EVALUATION DEFINITIONS AND REPORTING SCOPE

E.1 PROCESS EVALUATION AND CANDIDATE SELECTION

For Table 3, step-preference accuracy measures agreement between the scorer ordering and the reference preference for two steps with the same question and prefix. First-error accuracy requires the predicted first erroneous step to match the reference position; predicting no erroneous step is the explicit all-correct decision. The error threshold is selected on process-development data only. For an independently rated step with utility u_i, the absolute error is $|r_i^G - u_i|$; utility MAE summarizes these errors against evidence-support ratings rather than the training-rule targets.

For $T_{\text{parsed}} > 0$, Table 4 scores a candidate by $S(y) = T_{\text{parsed}}^{-1} \sum_{t=1}^{T_{\text{parsed}}} r_t^{\text{proc}}$ over its parsed steps. Invalid candidates, including wholly unparseable responses, rank last; original candidate indices break Top-1 selection ties. For a correct–wrong candidate pair, ranking receives one point when the correct candidate scores higher, zero when it scores lower, and one half when their scores tie. Top-1 EM applies the answer evaluator to the selected response and includes all questions; reference answers identify outcomes but do not enter the process score. Four candidates per question and identical scoring models are shared across allocations.

E.2 OPERATIONAL REQUIREMENTS FOR DIRECT-RULE SCORING

The direct-rule control requires source absence, a valid semantic abstention, and a technical scoring failure to be handled separately. Absence of an admissible source slot gives $m_t = 0$ and the masked textual fallback. A valid UNKNOWN judgment follows the annotation rules in Appendix B and cannot by itself establish a contradiction. A timeout, truncated or invalid annotation response, or failure to read a required cached asset does not establish zero utility. The required response is to suspend the affected batch before an optimizer update, retain the rollouts and failure record, and resolve or rescore the failed requests before continuing. Assigning zero or selectively dropping failed candidates would change the control. These are operational requirements, not measurements of failure frequency or recovery outcomes.

E.3 SCOPE OF THE REPORTED COMPARISONS

Table 1 compares reference systems; the matched variants in Table 2 provide the basis for training-objective attribution. The fixed-mean control tests whether step-dependent allocation improves on a training-average constant, not whether it outperforms a development-tuned constant. Likewise, independent evidence-support ratings and answer selection assess different properties of a process scorer; neither measurement alone establishes general semantic-verification ability.

F Comparison of process supervision methods

The comparison separates the object being evaluated, the source of supervision, and the role of process feedback in each method. It concerns methodological scope rather than comparative benchmark performance.

Table 9: Comparison of process supervision methods.

Method	Object	Supervision	Use
GraphPRM (Peng et al., 2025)	Graph-computation steps	Binary correctness from algorithms and MCTS	Inference search and preference training
DPRM (Wang et al., 2026)	CoT steps and KG triples	Outcome-derived implicit rewards	Iterative selection during inference
ReasonRAG (Zhang et al., 2025a)	Query, evidence, and answer actions	Answer-guided process preferences	Agentic RAG policy training
Q-PRM (Ye et al., 2025)	Query-rewriting steps	MCTS-derived step preferences	Self-training and PRM-guided decoding
SynerGate-RAG	Reader steps given passages and a KG	Explicit signed weak graph utility	Gated graph–text rewards for reader training

MCTS denotes Monte Carlo tree search. GraphPRM produces continuous scores despite binary training labels. SynerGate-RAG uses an availability mask and a learned gate to combine graph and text feedback; deployment requires no reward-model calls.

G PER-RUN ANSWER-QUALITY RESULTS

The per-run scores underlying Table 1 are listed below. For a metric x_s at evaluation seed s, the mean and sample standard deviation are

$$\bar{x} = \frac{1}{3} \sum_{s \in \{42,43,44\}} x_s, \qquad s_x = \sqrt{\frac{1}{2} \sum_{s \in \{42,43,44\}} (x_s - \bar{x})^2}.$$

For macro EM or F1, x_s is the unweighted mean over the three datasets in run s. Computation uses the dataset scores recorded to one decimal place and retains the derived macro precision until final rounding. Table 1 displays means. The standard deviations discussed in Section 4.2 are in percentage points, not standard errors or confidence intervals. These summaries quantify evaluation-run variation; they do not establish training-seed robustness or statistical significance.

Table 10: Per-run answer quality (EM/F1, %). Macro columns are recomputed from the three dataset scores.

Seed	Method	HotpotQA		2Wiki		MuSiQue		Macro	
		EM	F1	EM	F1	EM	F1	EM	F1
42	Zero-shot	0.3	4.0	0.0	4.3	0.0	3.0	0.1	3.8
	Naive RAG	24.7	36.7	4.0	18.0	8.0	20.3	12.2	25.0
	Self-RAG	16.3	27.3	10.3	24.0	8.3	21.0	11.6	24.1
	R1-Searcher	37.3	52.0	36.7	45.3	18.3	26.7	30.8	41.3
	CoRAG	21.3	28.7	18.3	28.3	12.7	18.3	17.4	25.1
	IRCoT	37.7	49.3	35.0	46.7	21.7	33.3	31.5	43.1
	ReaRAG	34.3	46.7	31.7	47.7	26.3	38.0	30.8	44.1
	SynerGate-RAG	45.0	60.0	63.0	70.0	28.7	44.0	45.6	58.0
43	Zero-shot	0.0	4.3	0.0	4.0	0.3	3.0	0.1	3.8
	Naive RAG	26.0	38.7	5.3	17.0	9.7	18.7	13.7	24.8
	Self-RAG	17.0	29.3	10.3	25.0	8.0	20.3	11.8	24.9
	R1-Searcher	39.7	51.7	38.3	44.3	16.7	25.7	31.6	40.6
	CoRAG	19.3	27.7	18.7	29.3	12.3	16.7	16.8	24.6
	IRCoT	38.0	50.0	34.7	45.7	22.0	35.0	31.6	43.6
	ReaRAG	33.7	47.0	30.3	47.0	25.3	40.3	29.8	44.8
	SynerGate-RAG	45.3	57.3	64.7	68.0	28.3	42.3	46.1	55.9
44	Zero-shot	0.0	4.0	0.3	4.3	0.0	3.0	0.1	3.8
	Naive RAG	25.7	37.3	6.3	16.3	10.0	20.0	14.0	24.5
	Self-RAG	18.3	30.0	11.0	24.3	10.0	20.0	13.1	24.8
	R1-Searcher	40.0	49.7	36.7	45.0	17.0	28.3	31.2	41.0
	CoRAG	20.0	29.3	18.3	29.3	12.7	19.0	17.0	25.9
	IRCoT	36.7	49.3	35.7	47.0	23.0	34.7	31.8	43.7
	ReaRAG	33.3	48.0	32.0	48.0	26.7	39.0	30.7	45.0
	SynerGate-RAG	44.7	58.3	63.7	68.0	29.3	41.3	45.9	55.9