ReCon: Toward Balanced Learning under Inter-Context and Context-Memory Conflicts

Huipeng Ma[1,2] **Dandan Song**[1,*] **Shaonan Ma**[2,*] **Changzhi Zhou**[1] **Jun Yang**[1]
Yuhang Tian[1] **Luan Zhang**[1] **Chenhao Li**[1] **Xudong Li**[1] **Fang Xi**[2] **Guangyuan Feng**[1]

[1]**Beijing Institute of Technology** [2]**Qiyuan Lab**

{mahuipeng,sdd}@bit.edu.cn mashaonan@qiyuanlab.com

Abstract

In retrieval-augmented generation (RAG), knowledge conflicts arise when retrieved contexts disagree with each other or with the model's parametric memory. Reinforcement Learning with Verifiable Rewards (RLVR) has emerged as a standard approach to address such conflicts. However, existing RLVR-based methods are typically formulated under a single-conflict assumption. In realistic mixed-conflict settings where Inter-Context and Context-Memory co-exist, unified RLVR optimization does not distinguish conflict types. As a result, the type with stronger advantage signals dominates policy updates, leaving the other under-optimized and thus yielding imbalanced performance across conflict types. To address this challenge, we propose **ReCon**, a type-aware RLVR framework for balanced learning across Inter-Context and Context-Memory conflicts. Specifically, ReCon computes a step-level imbalance signal from type-wise mean absolute advantage and triggers resampling on the weaker conflict type. Instead of resampling from scratch, ReCon branches from informative failures. It selects an informative failed trajectory using uncertainty and closeness signals, then branches at an entropy-jump point, concentrating additional rollouts on critical reasoning decisions. Experiments show that ReCon yields stronger and more balanced performance across knowledge-conflict and multi-hop QA benchmarks, and also generalizing well under mixed-conflict settings.

1 Introduction

Retrieval-augmented generation (RAG) introduces external contexts to large language models during reasoning, enabling them to exploit retrieved evidence in addition to their parametric knowledge [2, 10, 12, 14, 17, 34, 39]. Despite this, introducing external evidence forces the model to confront multiple sources of information that may conflict during reasoning. Making reliable decisions when confronted with such conflicting evidence has become one of the central challenges in understanding and improving the behavior of RAG models [3, 29].

Prior work has proposed various methods to handle knowledge conflicts. Several methods focus on Context-Memory conflicts, aiming to reconcile discrepancies between retrieved evidence and the model's internal knowledge[1, 6, 23]. Others investigate Inter-Context conflicts, studying how to mitigate disagreements across multiple retrieved contexts [3, 22, 24].

With the success of DeepSeek-R1 [8, 13], Reinforcement Learning with Verifiable Rewards (RLVR) [13, 16, 32, 36] has emerged as an effective post-training paradigm for improving the reasoning abilities of LLMs. Recent studies have also applied RLVR to knowledge-conflict settings and use verifiable feedback to model different conflict types in a targeted manner. For Inter-Context

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

conflicts, KCR [37] pairs contradictory contexts and optimizes verifiable rewards to encourage coherent selection. For Context-Memory conflicts, Trustworthy-Alignment [35] uses counterfactual contexts to align the model toward context-grounded answers, while Knowledgeable-R1 [15] leverages multiple specialized policies to decide when to follow context or rely on parametric knowledge.

However, these RLVR-based approaches are typically formulated under a single-conflict assumption. In practice, models often need to handle mixed-conflict scenarios where both conflict types co-exist. We therefore consider a more realistic mixed-conflict setting, where Inter-Context and Context-Memory conflicts co-exist and require joint optimization. Nevertheless, under RLVR optimization, naively mixing both conflict types in the training data is not sufficient. These RLVR pipelines typically apply the same advantage estimator to all conflict examples and aggregating them into a single update step. As a result, the training process does not explicitly account for differences in characteristics or difficulty between different types of conflicts. In this uniform treatment, conflict types that contribute more substantial advantage signals progressively dominate policy updates, whereas those with persistently weaker advantages are diluted during gradient aggregation and remain under-optimized, leading to a systematic performance imbalance across conflict types.

Motivated by this, we propose **ReCon**, which rebalances type-wise mean absolute advantages for balanced learning under mixed knowledge conflicts. Specifically, it first partitions training examples into Inter-Context and Context-Memory groups, and tracks the mean absolute advantage of each group during training. When a clear imbalance in type-wise mean absolute advantages is detected, ReCon treats the conflict type with the smaller mean absolute advantage as the weaker type. It then preferentially resamples examples from this type to amplify its effective learning signal and promote more balanced policy optimization across conflict types. Rather than resampling from scratch, this resampling is implemented by selecting an informative failed trajectory using uncertainty and closeness signals, with subsequent branching at an entropy-jump point along it, thereby allocating additional rollouts to critical reasoning decisions.

Experiments on four conflict and multi-hop QA benchmarks (ConflictBANK, RAMDocs, HotpotQA, 2WikiMultiHopQA), across three backbones spanning different scales and families (Qwen2.5-3B/7B-Instruct and Llama-3.1-8B-Instruct), show that ReCon yields stronger and more balanced performance over strong baselines, while generalizing well under mixed-conflict settings. Our contributions can be summarized as follows:

- We introduce ReCon for balanced learning under mixed knowledge conflicts, explicitly addressing the optimization imbalance between Inter-Context and Context-Memory conflicts.
- We design a sampling strategy within ReCon that resamples informative examples from the weaker conflict type to rebalance policy improvement.
- We conduct experiments on knowledge-conflict and multi-hop QA benchmarks, showing that ReCon yields stronger and more balanced performance across mixed-conflict settings.

2 Preliminary

2.1 Knowledge Conflict.

We consider context-conditioned question answering. Each sample is represented as $(q, \mathcal{C}, a^*)$, where q is the input question, $\mathcal{C} = \{c_1, \ldots, c_m\}$ is the associated context set, and a^* is the target answer. Different c_i may correspond to different candidate answers, thereby allowing $\mathcal{C}$ to contain multiple mutually inconsistent candidate answers (Inter-Context conflicts). Moreover, the conclusion implied by $\mathcal{C}$ may contradict the model's parametric memory (Context-Memory conflicts). Given $(q, \mathcal{C})$ and a prompt template p, the model π_θ generates an answer $y \sim \pi_\theta(\cdot \mid q, \mathcal{C}, p)$, and correctness is measured by the agreement between y and a^*.

2.2 RLVR and GRPO.

We follow the RLVR paradigm, which trains LLMs with rewards derived from deterministic verifiers rather than learned reward models. Given a generated answer y and a ground-truth answer a^*, a verifier $V(\cdot)$ returns a scalar reward $r = V(y, a^*)$ that reflects answer correctness.

For policy optimization, we employ GRPO, which removes the value model by replacing it with group-wise reward normalization and a KL-regularized clipped policy gradient. For an instance

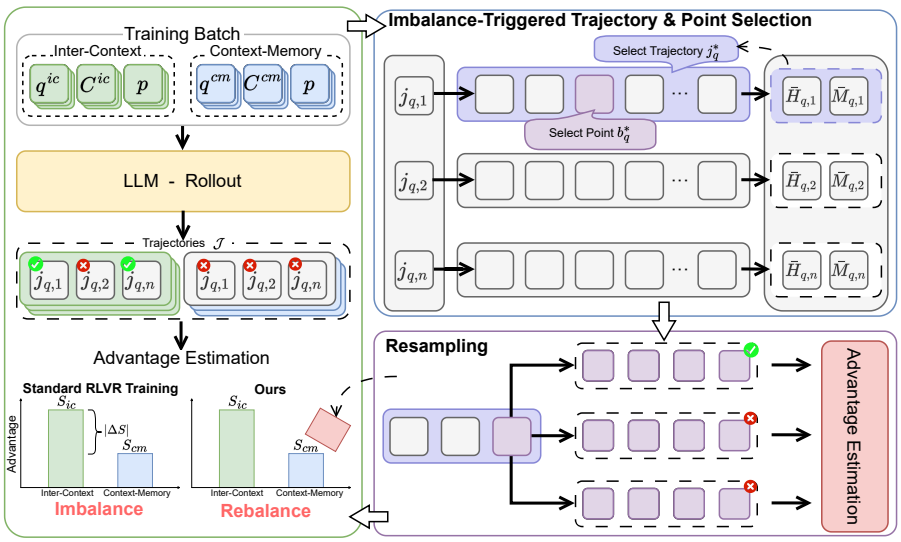

Figure 1: Overview of ReCon. Upon detecting type imbalance, it selects a branching trajectory and point, then resamples from that point to generate additional rollouts and update the policy.

$(q, \mathcal{C}, a^*)$, the policy π_θ generates G answers $\{y_i\}_{i=1}^G$ with verifier rewards $\{r_i\}_{i=1}^G$. Let μ_r and σ_r denote the group mean and standard deviation of $\{r_i\}$. GRPO computes the group-relative advantage

$$\hat{A}_i = (r_i - \mu_r)/(\sigma_r + \varepsilon), \tag{1}$$

where $\varepsilon > 0$ prevents division by zero. We define the importance ratio $\rho_i = \pi_\theta(y_i \mid q, \mathcal{C})/\pi_{\text{old}}(y_i \mid q, \mathcal{C})$, and denote $\text{CLIP} = \text{clip}(\rho_i, 1-\epsilon, 1+\epsilon)$. The GRPO objective is

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\,\mathbb{E}_i\Big[\min\big(\rho_i \hat{A}_i,\ \text{CLIP}\cdot\hat{A}_i\big)\Big] + \beta\,\mathbb{E}_i\Big[\text{KL}\big(\pi_\theta \,\|\, \pi_{\text{ref}}\big)\Big], \tag{2}$$

where π_{ref} is a fixed reference policy and β controls the KL regularization strength. Throughout the paper, we instantiate the verifier V as an exact-match check between y and a^*, so $r_i \in \{0, 1\}$.

3 Method

We propose **ReCon**, which incorporates Inter-Context and Context-Memory knowledge conflicts in a unified modeling and optimization process. As shown in Figure 1, ReCon continuously monitors type-wise mean absolute advantages during training to detect imbalance across conflict types. Once imbalance is observed, it selectively resamples more informative instances from the weaker type to rebalance policy improvement.

3.1 Step-level Imbalance Signal

We introduce a step-level statistic that quantifies the update strength contributed by each conflict type. Since GRPO scales policy updates by advantages, the mean absolute advantage over a subset of samples directly reflects the effective learning signal from that subset.

Type-wise mean absolute advantage. Let $\mathcal{T} = \{\text{ic}, \text{cm}\}$ denote the two conflict types. At each training step, we sample G rollouts per question in the current mini-batch, and let $A_{q,k}$ denote the GRPO advantage of the k-th rollout for question q with conflict-type label t_q. The type-wise cumulative advantage is

$$S_t \;=\; \frac{1}{G\,|\mathcal{Q}_t|} \sum_{q \in \mathcal{Q}_t} \sum_{k=1}^{G} \big|A_{q,k}\big|, \qquad t \in \mathcal{T}. \tag{3}$$

where $\mathcal{Q}_t = \{q : t_q = t\}$ is the set of questions of type t in the current mini-batch.

Imbalance-triggered weaker type. We define the step-level imbalance as $\Delta S = S_{\text{ic}} - S_{\text{cm}}$, and identify the weaker conflict type as the one with the smaller mean absolute advantage:

$$t^* \;=\; \arg\min_{t \in \mathcal{T}} S_t. \tag{4}$$

Resampling is triggered only when the imbalance is sufficiently pronounced, i.e., $|\Delta S| > \epsilon_{\text{stop}}$, where $\epsilon_{\text{stop}} > 0$ is a predefined threshold.

3.2 Imbalance-Triggered Resampling

Once imbalance is detected and the weaker type t^* is identified (Section 3.1), ReCon increases exploration on t^* by selectively resampling informative questions. Naive resampling from scratch, however, is of limited benefit when all rollouts fail. ReCon instead focuses the extra compute on the most informative part of the reasoning process, by branching from a high-uncertainty failed trajectory at a critical decision point that is more likely to alter the final outcome.

3.2.1 Failure buffer maintenance.

We maintain a failure buffer $\mathcal{F}_{t^*}$ of recent failed questions for the weaker type. A question q is inserted into $\mathcal{F}_{t^*}$ when all G rollouts fail the verifier in the current step, and is removed once it yields at least one successful rollout. To retain only recent failures, $\mathcal{F}_{t^*}$ is implemented as a sliding window of size W.

When resampling is triggered, ReCon uniformly samples R candidate questions from $\mathcal{F}_{t^*}$; if the buffer is empty, resampling is skipped for this step.

3.2.2 Trajectory selection.

For each candidate question q, we select a failed trajectory j_q^* from $\mathcal{J}_q = \{j_{q,k}\}_{k=1}^{G}$ based on two complementary signals: an *entropy-based uncertainty* signal that captures where the policy is unsure, and a *preference-based closeness* signal that captures whether the policy is close to recovering the ground-truth answer.

Entropy-based uncertainty signal. A failed trajectory that exposes *critical* reasoning decisions is more likely to yield informative branches when resampled. We capture this with the mean token-level entropy along the trajectory,

$$\bar{H}_{q,k} \;=\; \mathrm{Norm}_q\left(\frac{1}{L}\sum_{u=1}^{L} H_{q,k}(u)\right), \tag{5}$$

where $H_{q,k}(u)$ is the token entropy at position u of trajectory $j_{q,k}$, $L = |j_{q,k}|$ is its length in tokens, and $\mathrm{Norm}_q(\cdot)$ denotes min-max normalization across the G trajectories of question q.

Preference-based closeness signal. Entropy alone reflects uncertainty, but not whether the probability mass is actually moving toward the ground-truth answer. We therefore complement entropy with the policy's *relative preference* for the ground-truth answer over the generated one, evaluated under the same teacher-forcing prefix $\mathrm{pref}(q,k)$, i.e., the reasoning segment of trajectory $j_{q,k}$ that precedes the final answer. The closeness score is

$$\bar{M}_{q,k} \;=\; 1 - \mathrm{Norm}_q\big(\ell_\theta(a_q^*) - \ell_\theta(\tilde{a}_{q,k})\big), \tag{6}$$

where $\ell_\theta(a) = \frac{1}{|a|}\log p_\theta(a \mid q, \mathcal{C}, \mathrm{pref}(q,k))$ is the length-normalized log-probability of answer a, a_q^* is the ground-truth answer, and $\tilde{a}_{q,k}$ is the answer produced by trajectory $j_{q,k}$.

Trajectory selection. We combine the two signals into a joint score and select the trajectory with the highest score,

$$j_q^* \;=\; \arg\max_{k \in \{1,\ldots,G\}} \big[\, \alpha_e\, \bar{H}_{q,k} + (1-\alpha_e)\, \bar{M}_{q,k} \,\big], \tag{7}$$

where $\alpha_e \in [0,1]$ balances uncertainty and closeness.

3.2.3 Branching point selection.

Given the token-level entropy sequence $\{H_{q,*}(u)\}_{u=1}^{L_{q,*}}$ along the selected trajectory j_q^*, we branch at a position where uncertainty rises sharply, which often corresponds to a critical reasoning decision. We first take the top-K highest-entropy positions as candidates, denoted by $\mathcal{O}_K$. For each $s \in \mathcal{O}_K$, we compute a local entropy-jump score against its preceding context, and select the branching point with the largest jump,

$$b_q^* \;=\; \arg\max_{s \in \mathcal{O}_K} \left[H_{q,*}(s) - \frac{1}{\ell_s} \sum_{u=s-\ell_s}^{s-1} H_{q,*}(u) \right], \tag{8}$$

where $\ell_s = \min(w, s-1)$ is the length of the look-back window and w is a fixed window size.

3.3 Policy Optimization

At each RLVR step, we sample G rollouts per instance and compute GRPO advantages from verifier rewards. A verifiable correctness reward based on ACC is used, assigning a reward of 1 when the predicted answer matches the gold answer and 0 otherwise. We then aggregate advantages by conflict type to obtain $S_{\text{ic}}, S_{\text{cm}}$ and an imbalance signal ΔS; if $|\Delta S|$ is small we proceed with a standard GRPO update, otherwise we identify the weaker type t^* and trigger resampling. When resampling is triggered, we draw candidate questions from the failure buffer $\mathcal{F}_{t^*}$. For each candidate, ReCon branches from an informative failed trajectory selected by entropy and closeness signals, chooses a branching point by an entropy-jump criterion, and samples extra rollouts to augment the current-step data before performing the GRPO update.

4 Experimental Setup

4.1 Datasets

We evaluate ReCon in two settings. For knowledge-conflict benchmarks, we train on a subset of **ConflictBANK** [22] and evaluate on both **ConflictBANK** and **RAMDocs** [24]. ConflictBANK provides each question with supporting and conflicting evidence documents, making it suitable for constructing mixed-conflict training data. Based on it, we build two conflict settings, *Inter-Context* and *Context-Memory*, as described in Appendix A.1. We use RAMDocs as an additional benchmark to assess cross-benchmark generalization under more diverse sources of conflicting evidence. For multi-hop QA benchmarks, we further construct counterfactual passages to explicitly introduce knowledge conflicts. We train ReCon on a subset of **HotpotQA** [31] and evaluate on the validation sets of **HotpotQA** and **2WikiMultiHopQA** [11]. Details of the counterfactual construction are provided in Appendix A.2.

4.2 Implementation Details

All experiments are implemented in the VERL [21] framework with GRPO [8]. We use Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct [30], and Llama-3.1-8B-Instruct [7] as the backbones and run training on 8× NVIDIA A800 GPUs. We set the batch size to 256, max prompt to 2048, response lengths to 512, and optimize with learning rate 1×10^{-6}. For rollouts, we sample 5 responses with temperature 0.7, top-k 10, and top-p 0.95. We report micro-F1 score as the evaluation metric, which measures the overlap between the generated answer and the ground truth. Prompt templates are provided in Appendix C.

4.3 Baselines

We compare ReCon with several representative baselines. **CK-PLUG** [1] is a decoding-time method that detects knowledge conflicts via confidence gain and adjusts token probabilities to control parametric versus contextual reliance. **GRPO** [8] is a verifier-based RL baseline with group-wise reward normalization. **Trustworthy-Alignment** [35] and **Knowledgeable-R1** [15] are two representative methods for Context-Memory conflicts, respectively encouraging context-grounded answers and stronger reliance on parametric knowledge under misleading evidence. **KCR** [37] is an Inter-Context

Table 1: Main results (F1) on knowledge-conflict benchmarks and multi-hop QA benchmarks. For ConflictBANK and RAMDocs, IC and CM denote Inter-Context and Context-Memory conflicts, respectively, and ALL denotes the aggregate performance across the two conflict types. Best results within each model block are in bold.

Method	ConflictBANK			RAMDocs			Multi-hop QA	
	IC	CM	ALL	IC	CM	ALL	HotpotQA	2WikiMQA
Qwen2.5-3B-Instruct								
CK-PLUG	51.20	31.48	40.29	42.11	39.25	40.74	13.43	14.98
GRPO	68.70	54.62	64.97	50.86	36.00	44.92	20.44	23.82
Trustworthy-Alignment	63.79	50.03	55.24	41.53	30.79	35.11	22.75	26.21
Knowledgeable-R1	63.24	57.11	60.17	46.83	39.28	42.69	26.96	29.63
KCR	73.07	53.21	66.17	52.67	32.78	43.11	29.97	32.44
SeqGRPO (IC→CM)	61.90	56.00	58.14	43.54	35.33	38.86	20.16	25.64
SeqGRPO (CM→IC)	69.29	51.39	61.46	50.85	34.73	43.01	23.77	24.67
ReCon (Ours)	**75.11**	**63.45**	**71.96**	**54.72**	**44.93**	**51.78**	**31.67**	**34.21**
Qwen2.5-7B-Instruct								
CK-PLUG	63.23	47.70	56.11	42.33	39.05	40.54	18.74	19.76
GRPO	78.43	74.15	76.59	57.19	49.21	53.27	26.85	29.63
Trustworthy-Alignment	77.84	73.27	75.98	51.00	44.10	47.64	29.65	32.22
Knowledgeable-R1	74.83	78.24	76.35	56.60	52.27	54.26	31.54	31.43
KCR	**82.20**	74.60	79.33	**59.35**	46.11	52.31	35.17	34.96
SeqGRPO (IC→CM)	70.76	77.33	73.15	49.52	50.17	49.92	25.11	30.73
SeqGRPO (CM→IC)	79.74	69.24	75.09	58.21	47.65	52.66	28.77	31.13
ReCon (Ours)	82.03	**82.97**	**82.75**	59.21	**57.97**	**58.74**	**37.67**	**40.18**
Llama-3.1-8B-Instruct								
CK-PLUG	59.71	42.26	51.35	44.98	41.17	42.96	18.57	20.62
GRPO	74.36	70.45	72.59	53.79	48.06	51.73	25.15	27.87
Trustworthy-Alignment	76.93	69.41	73.51	53.01	43.65	48.82	32.65	35.13
Knowledgeable-R1	67.71	**77.92**	72.84	50.47	53.14	52.59	31.11	32.75
KCR	78.04	69.71	74.29	57.27	44.81	50.04	33.67	35.59
SeqGRPO (IC→CM)	69.95	74.13	72.19	50.01	48.56	49.86	24.97	28.78
SeqGRPO (CM→IC)	74.18	67.55	70.83	52.00	46.58	49.03	26.76	30.83
ReCon (Ours)	**79.69**	76.97	**77.03**	**58.57**	**54.14**	**56.66**	**35.26**	**38.35**

conflict baseline that organizes conflicting contexts into reasoning paths and optimizes logic-based rewards. We further include **SeqGRPO**, a sequential training variant of GRPO inspired by curriculum-style optimization [4, 19, 33], which trains on one conflict type first and then the other. We evaluate both orders, IC→CM and CM→IC, to test whether mixed conflict types can be handled effectively through sequential optimization.

5 Experimental Results

5.1 Main Results

The experimental results are presented in Table 1. We can obtain the following observations:

Competitive performance under mixed conflicts. Table 1 reports F1 results on ConflictBANK, RAMDocs, and two multi-hop QA benchmarks across Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct, and Llama-3.1-8B-Instruct. On ConflictBANK, ReCon consistently achieves the best overall performance across all three model scales. More importantly, the gains are consistent at the type level: ReCon improves both Inter-Context (IC) and Context-Memory (CM) scores rather than overfitting to a single type. In particular, ReCon remains competitive with the strongest type-specialized baseline on each conflict type (KCR on IC, Knowledgeable-R1 on CM), while consistently achieving the best overall score (ALL) across all three backbones.

Generalization across backbones and benchmarks. Beyond in-domain performance on ConflictBANK, ReCon also generalizes well across backbones and benchmarks. Across three backbones spanning different scales and model families (Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct, and Llama-3.1-8B-Instruct), ReCon consistently achieves the best overall performance, indicating that its type-aware rebalancing does not rely on a specific model size or pretraining distribution. On RAMDocs, ReCon (trained on ConflictBANK) maintains the best overall F1 with notable gains on CM, demonstrating cross-benchmark robustness under more diverse sources of conflicting evidence. A similar trend holds on the multi-hop QA benchmarks, where ReCon consistently outperforms strong baselines on both in-domain HotpotQA and the out-of-distribution 2WikiMQA.

Table 2: Ablation results on ConflictBANK, RAMDocs, HotpotQA, and 2WikiMQA using Qwen2.5-7B-Instruct, measured by F1. The upper section shows the effect of resampling, the middle section shows trajectory selection ablations, and the lower section shows branching point selection ablations.

Method	ConflictBANK			RAMDocs			Multi-hop QA	
	IC	CM	ALL	IC	CM	ALL	HotpotQA	2WikiMQA
GRPO	78.43	74.15	76.59	57.19	49.21	53.27	26.85	29.63
ReCon (Ours)	**82.03**	**82.97**	**82.75**	**59.21**	**57.97**	**58.74**	**37.67**	**40.18**
-*w* scratch	79.16	76.33	77.52	57.74	51.38	54.56	27.02	29.76
-*w* type-agnostic	**82.32**	74.21	77.83	–	–	–	–	–
-*w/o* uncertainty	81.94	79.37	80.92	58.24	55.78	57.03	35.45	38.31
-*w/o* closeness	79.94	77.36	78.00	57.86	52.00	55.21	29.17	32.74
-*w* random	79.98	78.36	78.73	58.00	52.79	55.34	30.06	33.91
-*w* top-entropy	80.11	79.37	79.68	57.08	55.16	56.25	33.83	35.88

Balanced improvements across conflict types. Table 1 reveals a clear imbalance pattern among existing type-specialized methods. KCR, which targets IC conflicts, excels on IC but degrades noticeably on CM; in contrast, Trustworthy-Alignment and Knowledgeable-R1, both designed for CM conflicts, exhibit the opposite trend. This indicates that type-specialized optimization only amplifies the model's response to one conflict signal, while the learning signal of the other conflict type is systematically suppressed, leading to a structural type bias. In contrast, ReCon does not optimize for a single conflict type; instead, it continuously monitors the type-wise mean absolute advantages of both conflicts during training and triggers resampling on the weaker type once imbalance is detected, thereby allocating additional rollout budget where it is most needed. This mechanism-level rebalancing preserves the learning signals of both IC and CM conflicts, yielding stable improvements on both types simultaneously rather than trading one off against the other.

Sequential training causes type forgetting. A natural question is whether IC and CM conflicts can be addressed via curriculum-style sequential training, where the model is first trained on one conflict type until convergence and then on the other. The two variants of SeqGRPO (IC→CM and CM→IC) directly test this hypothesis. As shown in Table 1, regardless of the training order, the conflict type trained in the first stage consistently suffers a clear performance drop in the second stage, even falling below the single-stage GRPO baseline. This indicates that IC and CM are not related by difficulty level but exhibit a degree of objective interference, analogous to catastrophic forgetting in multi-task learning. This finding directly motivates the central design of ReCon: the two conflict types must be jointly rebalanced within a single optimization process rather than handled sequentially. ReCon substantially outperforms both SeqGRPO variants, showing that type-aware online rebalancing handles mixed-conflict scenarios more effectively than any static training schedule.

5.2 Analysis of Resampling

We analyze the effect of resampling by comparing ReCon with an alternative from-scratch resampling approach. After identifying the weaker conflict type, this approach increases the rollout compute by drawing additional full rollouts from scratch under the same input, without any trajectory- or point-level guidance. As shown in the upper part of Table 2, from-scratch resampling yields limited improvements on the weaker type. A key reason is that, in the all-failure regime, simply drawing more full rollouts from the initial prompt has a low probability of reaching the correct answer. By contrast, ReCon branches from informative failures and concentrates additional rollouts around critical decisions, which more directly increases the chance of correcting the reasoning and produces stronger learning signals for the weaker type.

We also evaluate a type-agnostic resampler. It uses the same resampling procedure as ReCon but draws failures from a unified IC–CM buffer without identifying the weaker type. It therefore resamples at every step, with the per-step budget adjusted so that the total number of additional rollouts matches ReCon. On ConflictBANK, this variant is comparable to ReCon on IC but falls far behind on CM, widening the IC–CM gap from 0.94 to 8.11 points. A likely reason is that the model is already stronger on IC, so IC failures are easier to repair, and type-agnostic branching keeps reinforcing IC while leaving CM under-optimized. This shows that the gains of ReCon do not come from additional branching alone; deciding which conflict type to explore is also essential.

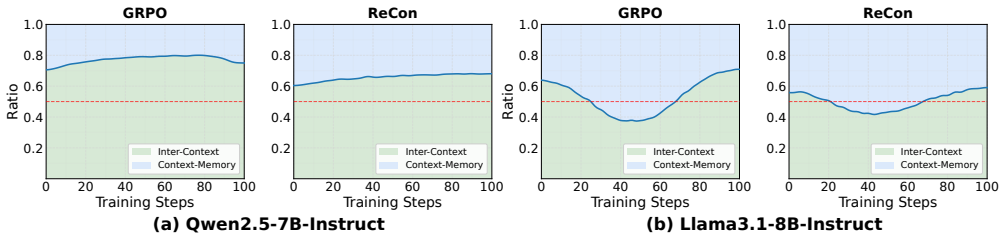

Figure 2: Type-wise cumulative-advantage ratio over training steps on ConflictBANK. The red dashed line (0.5) marks perfect balance between the two conflict types.

We further examine how optimization effort is distributed across conflict types during training. At each step, we measure the IC ratio $S_{\text{ic}}/(S_{\text{ic}} + S_{\text{cm}})$ and visualize it over training for both GRPO and ReCon on Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct. As shown in Figure 2, GRPO exhibits clear type-wise imbalance throughout training, but the manifestation differs across backbones: on Qwen2.5-7B-Instruct the ratio remains stably above 0.5 and gradually drifts further from balance, while on Llama-3.1-8B-Instruct it varies substantially, dropping below 0.5 before rebounding. In contrast, ReCon maintains a markedly more balanced and stable ratio across both backbones, staying closer to the 0.5 reference with much less variation. These results support our motivation that, without type-aware control, mixed-conflict optimization is systematically biased toward whichever conflict type produces stronger advantage signals, manifesting as either persistent dominance or unstable oscillation depending on the backbone. ReCon's imbalance-triggered resampling effectively mitigates both failure modes in a sustained manner.

5.3 Analysis of Trajectory Selection

We conduct an ablation study to analyze trajectory selection in ReCon. We examine how an entropy-based uncertainty signal interacts with a preference-based closeness signal. As shown in the middle part of Table 2, ReCon achieves the strongest and most consistent improvements when both signals are enabled, suggesting that the two signals work well together. The ablations also show that the two signals do not contribute equally. Using the preference-based closeness signal alone still retains non-trivial gains. This suggests that selecting trajectories closer to the desired answer behavior is already a strong criterion for effective correction. In contrast, using the entropy-based uncertainty signal alone yields smaller gains than using the preference-based closeness signal alone, suggesting that uncertainty by itself is a weaker indicator of which trajectories are correctable under mixed conflicts. These results suggest a clear synergy between the two signals. Closeness provides the primary guidance on which trajectory to fix, while entropy acts as an auxiliary cue that refines where to intervene when combined with closeness. Together, they enable more efficient and sustained rebalancing.

To corroborate the above ablation findings from a sampling-centric perspective, we report Pass@K accuracy on ConflictBANK with Qwen2.5-7B-Instruct. As shown in Figure 3(a), ReCon achieves the highest overall Pass@5 among all ablated settings, showing that its trajectory selection strategy translates into more successful sampling outcomes. Figure 3(b) further tracks Pass@5 as training progresses. While all methods improve over time, ReCon rises more rapidly and maintains a widening margin in later steps. Notably, closeness-only grows faster than uncertainty-only, while scratch remains consistently lowest, mirroring the ranking observed in the ablation table. Together, these curves suggest that ReCon's advantage is not merely an end-point effect; rather, it consistently improves the probability of sampling a correct trajectory during training, indicating a more effective use of the rollout budget.

5.4 Analysis of Branching Point Selection

We then investigate the impact of different branching point selection strategies in ReCon. After selecting an informative failed trajectory, ReCon branches at an entropy-jump point. We compare this strategy with two alternatives: *random*, which branches from a randomly chosen point on the selected trajectory, and *top-entropy*, which branches from the point with the highest token entropy. As shown in the bottom part of Table 2, both alternatives perform worse than the full method. Random

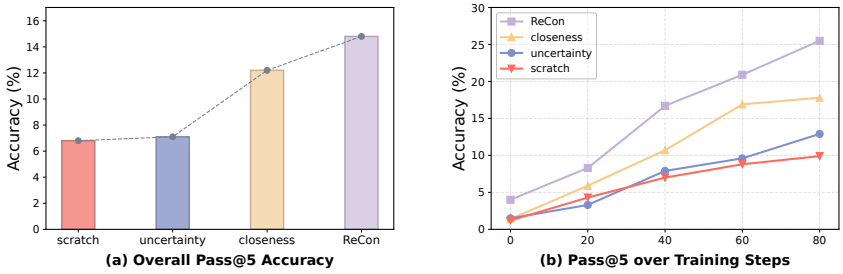

Figure 3: Pass@5 accuracy on ConflictBANK with Qwen2.5-7B-Instruct for scratch, uncertainty-only, closeness-only, and ReCon. (a) Overall Pass@5 accuracy. (b) Pass@5 over training steps.

branching often selects positions unrelated to the key reasoning error, leading to limited corrective benefit. Branching at the highest-entropy point performs better, indicating that uncertainty is a useful signal for intervention. However, it still underperforms ReCon, suggesting that high entropy alone is insufficient for identifying the most effective branching location. We attribute this to the fact that the highest-entropy point may reflect local ambiguity rather than a true reasoning transition. In contrast, the entropy-jump criterion used in ReCon favors positions where uncertainty rises sharply relative to the preceding context, making them more likely to correspond to critical reasoning decisions. This allows ReCon to allocate additional rollouts more effectively and achieve stronger gains.

6 Related Work

6.1 Knowledge Conflicts

Knowledge Conflict in LLMs. Recent work has emphasized that LLMs often face knowledge conflicts, where incompatible information can originate from the model's parametric memory or from retrieved external evidence [10, 12, 14, 18]. A recent survey [29] organizes knowledge conflicts into Intra-Context, Inter-Context, and Context-Memory conflicts, and highlights the limitations of existing mitigation approaches. **For Inter-Context conflicts**, prior work [3] calibrates the model to abstain when retrieved passages contain conflicting answer candidates, rather than forcing a single committed prediction. MADAM-RAG [24] mitigates Inter-Context conflicts by delegating each retrieved document to a separate agent for multi-round debate and then synthesizing their deliberations into a final answer. **For Context-Memory conflicts**, RARR [6] retrieves evidence after an initial response and performs evidence-grounded revision to better follow the retrieved context. Adacad [23] estimates the model's context sensitivity via the distributional discrepancy between decoding with and without context, and uses it to adaptively steer contrastive context-aware decoding. CK-PLUG [1] detects parameter–context disagreement using a Confidence Gain signal and adjusts token probabilities at decoding time to modulate reliance between parametric knowledge and retrieved evidence.

RLVR for Knowledge Conflicts. Motivated by recent progress in verifiable-reward training for reasoning models such as DeepSeek-R1 [8], Reinforcement Learning with Verifiable Rewards (RLVR) [13] has become a widely adopted post-training approach to enhance LLM reasoning. More recently, RLVR-based methods have been studied for resolving knowledge conflicts in LLMs [15, 35, 37]. **For Inter-Context conflicts**, KCR [37] pairs two conflicting contexts, extracts structured reasoning paths, and optimizes logic-style rewards to encourage selecting one coherent side rather than blending contradictions. **For Context-Memory conflicts**, Trustworthy-Alignment [35] constructs counterfactual contexts and trains the model to prefer the substituted-context answer over its original parametric answer, while Knowledgeable-R1 [15] trains three specialized policies and uses cross-policy advantage comparison to decide when to follow context versus fall back to parametric knowledge under misleading evidence. Although effective, existing RLVR-based methods often treat mixed conflict types uniformly, causing stronger-signal types to dominate optimization and leaving weaker types under-trained.

6.2 Imbalance and Exploration in RL for LLMs

Imbalanced Optimization in Multi-Task RL. Recent work shows that jointly optimizing heterogeneous data in RL post-training can bias updates toward certain tasks. Wu et al. [27] systematically analyze multi-task RL post-training and find that some tasks produce much larger gradients and dominate optimization, even though these tasks do not yield larger learning gains. They also observe that absolute advantage tracks gradient magnitude within a task but not across heterogeneous tasks, and they call for future corrections without proposing a concrete method. Several methods address imbalance by reweighting. DISCO [38] scales rewards according to domain frequency and prompt-level self-consistency, and MT-GRPO [20] adapts task weights to improve worst-task performance.

Entropy-Guided Exploration in RLVR. Token entropy has been widely used to locate critical decisions in reasoning. Wang et al. [25] find that a small fraction of high-entropy tokens act as forks in reasoning paths, and restrict policy-gradient updates to these tokens. In agentic RL, ARPO [5] uses an entropy-based adaptive rollout mechanism that branches sampling at high-entropy tool-call steps. ReCon also uses entropy, but for a different purpose: it selects where to branch additional rollouts from failed trajectories. Rather than choosing the token with the highest absolute entropy, ReCon selects the position with the largest local entropy increase among high-entropy candidates. Moreover, branching in ReCon is triggered only for the weaker conflict type, so the additional exploration is driven by type-wise imbalance rather than by entropy alone.

7 Conclusion

We study RLVR under mixed knowledge conflicts and identify a previously overlooked failure mode: uniformly optimizing Inter-Context and Context-Memory conflicts produces type-biased updates, where the conflict type with stronger advantage signals dominates while the weaker type remains under-trained. To address this, we propose **ReCon**, a conflict-type-aware framework that monitors type-wise mean absolute advantages, detects imbalance, and resamples from the weaker type to rebalance optimization. Specifically, ReCon branches from informative failures by selecting trajectories using uncertainty and closeness signals, and branches at entropy-jump points within those trajectories. Experiments on four benchmarks (ConflictBANK, RAMDocs, HotpotQA, and 2WikiMQA) across three model scales (Qwen2.5-3B-Instruct, Qwen2.5-7B- Instruct and Llama-3.1-8B-Instruct) show consistent improvements over strong baselines.

Limitations

ReCon relies on reliable conflict-type information to compute type-wise mean absolute advantages and trigger resampling, so its effectiveness may degrade when type signals are missing or noisy. It also introduces extra computation and training-time overhead from imbalance detection, trajectory scoring, and branching-point selection, which can be more noticeable for long reasoning traces and large models. Finally, our empirical validation is centered on knowledge-conflict and multi-hop QA benchmarks under an RLVR setting, and extending ReCon to other conflict taxonomies or retrieval-in-the-loop RAG training remains future work.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (Grant No. 2025YFB3003700) and the National Natural Science Foundation of China (Grant No. 62476025).

References

[1] Baolong Bi, Shenghua Liu, Yiwei Wang, Yilong Xu, Junfeng Fang, Lingrui Mei, and Xueqi Cheng. Parameters vs. context: Fine-grained control of knowledge reliance in language models. *arXiv preprint arXiv:2503.15888*, 2025.

[2] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR, 2022.

[3] Hung-Ting Chen, Michael Zhang, and Eunsol Choi. Rich knowledge sources bring complex knowledge conflicts: Recalibrating models to reflect conflicting evidence. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2292–2307, 2022.

[4] Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamalloo. Self-evolving curriculum for llm reasoning. *arXiv preprint arXiv:2505.14970*, 2025.

[5] Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. Agentic reinforced policy optimization. In *International Conference on Learning Representations*, 2026.

[6] Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. Rarr: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, 2023.

[7] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[8] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[9] Wangzhen Guo, Qinkang Gong, Yanghui Rao, and Hanjiang Lai. Counterfactual multihop qa: A cause-effect approach for reducing disconnected reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4214–4226, 2023.

[10] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR, 2020.

[11] 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*, pages 6609–6625, 2020.

[12] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume*, pages 874–880, 2021.

[13] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

[14] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

[15] Chenyu Lin, Yilin Wen, Du Su, Hexiang Tan, Fei Sun, Muhan Chen, Chenfu Bao, and Zhonghou Lyu. Resisting contextual interference in rag via parametric-knowledge reinforcement. *arXiv preprint arXiv:2506.05154*, 2025.

[16] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

[17] Huipeng Ma, Luan Zhang, Dandan Song, Linmei Hu, Yuhang Tian, Jun Yang, Changzhi Zhou, Chenhao Li, Yizhou Jin, Xudong Li, Meng Lin, Mingxing Zhang, and Shuhao Zhang. Actishade: Activating overshadowed knowledge to guide multi-hop reasoning in large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(38):32419–32427, Mar. 2026.

[18] Zijie Meng, Jiwen Liu, Yufei Liu, Chengzhuo Tong, Xiaoqiang Liu, Yuanxing Zhang, Yulong Xu, and Pengfei Wan. Argus: Stacked multi-view identity mosaic injection for subject-preserving video generation. *arXiv preprint arXiv:2606.11670*, 2026.

[19] Shubham Parashar, Shurui Gui, Xiner Li, Hongyi Ling, Sushil Vemuri, Blake Olson, Eric Li, Yu Zhang, James Caverlee, Dileep Kalathil, et al. Curriculum reinforcement learning from easy to hard tasks improves llm reasoning. *arXiv preprint arXiv:2506.06632*, 2025.

[20] Shyam Sundhar Ramesh, Xiaotong Ji, Matthieu Zimmer, Sangwoong Yoon, Zhiyong Wang, Haitham Bou Ammar, Aurelien Lucchi, and Ilija Bogunovic. Multi-task GRPO: Reliable LLM reasoning across tasks. In *International Conference on Machine Learning*, 2026.

[21] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297, 2025.

[22] Zhaochen Su, Jun Zhang, Xiaoye Qu, Tong Zhu, Yanshu Li, Jiashuo Sun, Juntao Li, Min Zhang, and Yu Cheng. Conflictbank: A benchmark for evaluating the influence of knowledge conflicts in llms. In *NeurIPS*, 2024.

[23] Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Adacad: Adaptively decoding to balance conflicts between contextual and parametric knowledge. 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)*, pages 11636–11652, 2025.

[24] Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Retrieval-augmented generation with conflicting evidence. *arXiv preprint arXiv:2504.13079*, 2025.

[25] Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *Advances in Neural Information Processing Systems*, 2025.

[26] Jian Wu, Linyi Yang, Zhen Wang, Manabu Okumura, and Yue Zhang. Cofca: A step-wise counterfactual multi-hop qa benchmark. *arXiv preprint arXiv:2402.11924*, 2024.

[27] Runzhe Wu, Ankur Samanta, Ayush Jain, Scott Fujimoto, Jeongyeol Kwon, Ben Kretzu, Youliang Yu, Kaveh Hassani, Boris Vidolov, and Yonathan Efroni. Imbalanced gradients in RL post-training of multi-task LLMs. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 3137–3150, Rabat, Morocco, 2026. Association for Computational Linguistics.

[28] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024.

[29] Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. Knowledge conflicts for llms: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8541–8565, 2024.

[30] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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

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

[33] Shijie Zhang, Guohao Sun, Kevin Zhang, Xiang Guo, and Rujun Guo. Clpo: Curriculum learning meets policy optimization for llm reasoning. *arXiv preprint arXiv:2509.25004*, 2025.

[34] Zhebin Zhang, Xinyu Zhang, Yuanhang Ren, Saijiang Shi, Meng Han, Yongkang Wu, Ruofei Lai, and Zhao Cao. Iag: Induction-augmented generation framework for answering reasoning questions. *arXiv preprint arXiv:2311.18397*, 2023.

[35] Zongmeng Zhang, Yufeng Shi, Jinhua Zhu, Wengang Zhou, Xiang Qi, Peng Zhang, and Houqiang Li. Trustworthy alignment of retrieval-augmented large language models via reinforcement learning. *arXiv preprint arXiv:2410.16843*, 2024.

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

[37] Xianda Zheng, Zijian Huang, Meng-Fen Chiang, Michael J Witbrock, and Kaiqi Zhao. Kcr: Resolving long-context knowledge conflicts via reasoning in llms. *arXiv preprint arXiv:2508.01273*, 2025.

[38] Yuhang Zhou, Jing Zhu, Shengyi Qian, Zhuokai Zhao, Xiyao Wang, Xiaoyu Liu, Ming Li, Paiheng Xu, Wei Ai, and Furong Huang. DISCO balances the scales: Adaptive domain- and difficulty-aware reinforcement learning on imbalanced data. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025.

[39] Jiaqi Zhu, Naili Xing, Hexiang Pan, Haotian Gao, Jianwei Yin, Xiaokui Xiao, and Beng Chin Ooi. Logictree-rag: Logic tree-guided retrieval-augmented generation for long-form patent drafting. *arXiv preprint arXiv:2609.30943*, 2026.

A Construction of the Dataset

The construction procedure is model-specific. Since the subsequent IC/CM assignment depends on memory-state probing, different backbone models may produce different no-context predictions for the same question due to differences in parametric knowledge and reasoning ability. As a result, the constructed datasets are generated separately for each model.

A.1 Construction from ConflictBANK

We build our two conflict settings by re-constructing contexts from ConflictBANK. In ConflictBANK, each QA instance is paired with one default evidence and three types of conflict evidence: misinformation conflict evidence, temporal conflict evidence, and semantic conflict evidence. Following ConflictBANK, misinformation conflicts correspond to inaccurate information, temporal conflicts arise from knowledge changes over time, and semantic conflicts stem from the polysemic nature of language. In our implementation, these four evidence fields are stored as `correct` (default evidence), `fact_conflict` (misinformation conflict evidence), `temporal_conflict` (temporal conflict evidence), and `semantic_conflict` (semantic conflict evidence).

Memory-state probing (no-context answering). To explicitly characterize the alignment between the model's parametric memory and the gold answer, we first query the LLM with *only* the question q (without any external documents) and obtain an initial prediction $\hat{a}$. We use the prompt shown in Figure 4. We use the correctness of this no-context prediction, i.e., whether $\hat{a} = a^\star$, to determine the subsequent context construction rule for this instance.

Document sampling and bucket definition. For each instance, we sample in total 1–4 documents and concatenate them into a single context $\mathcal{C}$. Let c be the number of supporting documents sampled from `correct`, and w be the number of conflicting documents sampled from the union of `fact_conflict`, `temporal_conflict`, and `semantic_conflict`. We denote each constructed context by a bucket (c, w) and form two settings as follows.

Case A: correct no-context prediction ($\hat{a} = a^\star$). In this case, the model's parametric memory agrees with the gold answer. We construct:

- **Inter-Context (IC).** We include both supporting and conflicting evidence so that contradictions occur *within* the provided context:
$$(c, w) \in \{(1, 1), (1, 2), (1, 3)\}.$$
- **Context-Memory (CM).** We provide only conflicting evidence, making the context directly oppose the model's memory:
$$(c, w) \in \{(0, 1), (0, 2), (0, 3)\}.$$

Case B: incorrect no-context prediction ($\hat{a} \neq a^\star$). In this case, the model's parametric memory tends to the wrong answer. To ensure that the context can correct this memory error, we *always* include one supporting document (i.e., $c = 1$) and sample $w \in \{0, 1, 2, 3\}$ conflicting documents:

- **Context-Memory (CM).** We provide only supporting evidence to correct the model's memory:
$$(c, w) = (1, 0).$$
- **Inter-Context (IC).** We add conflicting evidence on top of the supporting one, creating within-context contradictions and testing evidence selection:
$$(c, w) \in \{(1, 1), (1, 2), (1, 3)\}.$$

A.2 Construction from HotpotQA

For multi-hop QA benchmarks, we construct mixed-conflict examples based on HotpotQA. Each constructed instance contains five documents. To determine whether an instance should be used for Inter-Context (IC) or Context-Memory (CM) construction, we follow Appendix A.1 and first perform memory-state probing by querying the LLM with only the question q, without any external

Prompt 1

You are a helpful assistant. After the user asks a question, you first think carefully and then give the answer.
When responding, please keep the following points in mind:
- The reasoning process and answer are enclosed within <think> </think> and <answer> </answer> tags, respectively.
- Output your final answer directly between the tag <answer> </answer> without any intermediate steps.
Here is an example:
Question: What is the capital of China?
<think> reasoning process here </think>
<answer> Beijing </answer>
Now your question are as follows:

Figure 4: Prompt template for Memory-state probing.

documents, and obtain a no-context prediction $\hat{a}$. We then use whether $\hat{a} = a^\star$ to determine the subsequent construction rule for this instance.

Inter-Context (IC). For questions that the model can already answer correctly without context, we construct within-context conflicts while keeping the model's parametric knowledge aligned with the gold answer. Each IC instance contains five documents: one first-hop supporting document, one second-hop supporting document, one second-hop conflicting document, and two irrelevant documents. The second-hop conflicting document is constructed by replacing the final answer entity in the original second-hop supporting document with a counterfactual entity, while preserving the surrounding context and sentence structure as much as possible. Similar counterfactual construction has also been explored in prior work on multi-hop QA [9, 26]. This yields a document that remains locally natural but supports an incorrect reasoning path. In this way, the context includes both a correct reasoning path and a conflicting alternative, creating contradictions within the provided evidence.

Context-Memory (CM). For questions that the model answers incorrectly without context, we construct contexts that correct the model's parametric memory rather than introduce within-context contradictions. Each CM instance contains five documents: one first-hop supporting document, one second-hop supporting document, and three irrelevant documents. In this setting, the context consistently supports the gold answer, while the conflict arises between the model's incorrect parametric knowledge and the contextual evidence.

Following this construction, IC samples evaluate the model's ability to resolve contradictions among retrieved documents, whereas CM samples evaluate whether the model can revise incorrect parametric knowledge using supporting context. This allows us to study mixed knowledge conflicts in multi-hop QA under a unified document format.

B Evaluation Metric

We report **micro-F1** as the main evaluation metric for ConflictBANK and RAMDocs. For a prediction $\hat{y}$ and a gold answer $y^\star$, F1 is computed based on the token-level overlap between the two answers. Specifically, let P, R, and $F1$ denote precision, recall, and F1, respectively:

$$P = \frac{|\hat{y} \cap y^\star|}{|\hat{y}|}, \qquad R = \frac{|\hat{y} \cap y^\star|}{|y^\star|}, \qquad F1 = \frac{2PR}{P+R}.$$

For the **IC** and **CM** columns, we compute F1 separately on the Inter-Context and Context-Memory subsets. In contrast, the **ALL** score is computed as *micro-F1* over the union of all IC and CM examples, rather than as the arithmetic mean of the two subset-level scores. Formally, let $\mathcal{D}_{IC}$ and $\mathcal{D}_{CM}$ denote the IC and CM subsets. Then

$$\text{ALL} = \text{micro-F1}(\mathcal{D}_{IC} \cup \mathcal{D}_{CM}).$$

Prompt 2

You are an expert in retrieval question answering. You will be given a question and multiple documents that may be unreliable and, in extreme cases, entirely incorrect.
Your tasks:
- If the provided documents contain a reliable source, answer based on it; if the documents are unreliable and you have high-confidence internal knowledge about the question, answer using that knowledge.
- Provide your reasoning process between <think></think> and output the final answer between <answer></answer>.
Output rules:
- Put all reasoning only inside <think></think>;
- Put the final answer only inside <answer></answer>;
- Do not output anything outside these two tags.
Here is an example:
Documents:
{{Documents for your reference}}
Question: What is the capital of China?
<think> reasoning process here </think>
<answer> Beijing </answer>
Now your question and documents are as follows:

Figure 5: Prompt template for ReCon.

C Prompt Template

We adopt a single prompt template throughout training and evaluation. The full template is shown in Figure 5.

D Details of LLMs

The details of selected LLMs are as follows:

Llama3 [7] is a collection of pre-trained and instruction-tuned LLMs in 8 and 70B sizes. The instruction-tuned LLMs, called Llama-3.1-Instruct, are optimized for dialogue use cases. We selected **Llama-3.1-8B-Instruct**.

Qwen2.5 [30] is a series of Qwen large language models, including both base and instruction-tuned versions, spanning a parameter scale from 0.5 billion to 72 billion. We selected **Qwen2.5-3B-Instruct** and **Qwen2.5-7B-Instruct**.

E Baseline Implementation Details

All baselines, including Trustworthy-Alignment [**?**], KCR [**?**], and Knowledgeable-R1 [**?**], are retrained from scratch on the same mixed IC–CM training data, backbone models, and evaluation splits as ReCon. We do not use their original checkpoints, training data, or reported results. Below we describe the training configuration of each baseline and how it is adapted to IC and CM examples.

Trustworthy-Alignment. We keep its original PPO-based optimization and context-following reward. The actor and critic learning rates are 9.65×10^{-6} and 1×10^{-6}, respectively. We use a global batch size of 8, policy and value clipping coefficients of 0.2, $\gamma = 1.0$, $\lambda = 0.95$, and 10 PPO epochs per batch. The reward coefficients follow the original setting: $\alpha_0 = 3$, $\alpha_1 = 1$, $\alpha_2 = 0.05$, and $\alpha_3 = 2$, which correspond to the positive/negative answer reward, the unmatched-answer penalty, the KL penalty, and the generation-collapse penalty, respectively. The rollout temperature is linearly annealed from 2.0 to approximately 0, and the maximum prompt and response lengths are 2048 and 512. To adapt the reward to our data, we define the positive and negative answers according to the conflict type. For CM examples, the answer supported by the retrieved context is positive, and the conflicting parametric-memory answer is negative. For IC examples, the answer supported by the

Table 3: Sensitivity to the failure buffer window W on ConflictBANK using Qwen2.5-7B-Instruct. Here, W denotes the number of recent batches whose failed samples are retained in the buffer for resampling.

W	1	2	4	8	16
F1	81.26	82.06	**82.75**	80.92	79.78

reliable context is positive, and the answer supported by the conflicting context is negative. Each response receives a positive, negative, or unmatched reward depending on which answer it contains.

KCR. We keep its original GRPO-based optimization and reasoning-path-based reward. We use a learning rate of 2×10^{-6}, Adam with $\beta_1 = 0.9$ and $\beta_2 = 0.99$, a weight decay of 0.1, a warmup ratio of 0.1, a cosine learning-rate schedule, eight generations per prompt, a maximum gradient norm of 0.1, and 10 training epochs. The maximum sequence length and completion length are 2048 and 512. The logic-coherence, logic-consistency, answer-correctness, and format rewards are weighted by 2, 1, 2, and 1, respectively. Reference and generated reasoning paths are encoded with `bge-large-en-v1.5` [28]. For IC examples, evidence from the reliable context defines the correct side, and evidence from the conflicting context defines the conflicting side. For CM examples, evidence from the retrieved context defines the correct side, while the parametric-memory answer and its supporting evidence define the conflicting side. Based on these definitions, KCR rewards reasoning coherence, reasoning–answer consistency, and answer correctness.

Knowledgeable-R1. We keep its original joint optimization over the PK, CK, and RPK objectives. We use a learning rate of 1×10^{-6}, a global batch size of 128, a rollout temperature of 1.0, 16 PK and 16 CK trajectories per example, maximum prompt and response lengths of 2048, and 10 training epochs. The PK, CK, and RPK objectives are equally weighted, and the adaptive coefficient β is constrained to $[0.01, 1]$. For IC examples, the context input contains both reliable and conflicting documents. For CM examples, it contains the retrieved context that conflicts with the model's parametric knowledge. In both cases, the query-only branch contains only the question, and all trajectories are evaluated against the same ground-truth answer.

F Training Cost Analysis

We compare the training cost of ReCon and GRPO on ConflictBANK with Qwen2.5-7B-Instruct. GRPO requires 7.3 hours of training, while ReCon takes 8.6 hours, a 17.8% overhead, comparable to or smaller than the overhead of related GRPO-based methods such as Knowledgeable-R1 (24% over their GRPO baseline). This extra cost is accompanied by a clear performance gain: ReCon improves the overall F1 score from 76.59% to 82.75%, yielding a 6.16-point absolute improvement over GRPO. The additional overhead mainly comes from the extra rollout stage triggered after imbalance detection. Importantly, the gains are not due to uniformly increasing rollouts. As shown in our resampling ablation in Table 2, uniformly drawing additional rollouts from scratch (*w scratch*) yields much smaller gains than ReCon's targeted resampling, confirming that the extra budget is effective only when focused on the right cases.

G Sensitivity Analysis

We report additional sensitivity studies on three key hyperparameters in ReCon: the failure buffer window size W, the uncertainty–closeness weighting factor α_e, and the trigger threshold ϵ_{stop}.

G.1 Sensitivity to the Failure Buffer Window Size

We first analyze the sensitivity of ReCon to the failure buffer window size W. ReCon maintains a failure buffer for the weaker conflict type and stores failed samples for later resampling. Here, W denotes the number of recent batches whose failed samples are retained in the buffer. It therefore determines the temporal range from which resampling candidates are drawn.

Table 4: Sensitivity to the uncertainty–closeness weight α_e on ConflictBANK using Qwen2.5-7B-Instruct.

α_e	0	0.2	0.4	0.6	0.8	1.0
F1	80.92	81.38	**82.75**	80.26	78.96	78.00

Table 5: Sensitivity to the trigger threshold ϵ_{stop} on ConflictBANK using Qwen2.5-7B-Instruct. Trigger rate denotes the percentage of training steps in which resampling is activated.

ϵ_{stop}	3×10^{-3}	4×10^{-3}	5×10^{-3}	6×10^{-3}	7×10^{-3}
F1	82.46	82.37	**82.75**	82.11	81.89
Trigger Rate (%)	90.68	90.24	89.65	89.11	88.75

As shown in Table 3, ReCon remains relatively stable across a range of W values, suggesting that the method is not overly sensitive to the exact choice of this window. When W is too small, the buffer contains failures from only a few recent batches, which limits candidate diversity and may cause resampling to focus too narrowly on a small subset of instances. In contrast, when W is too large, the buffer may retain stale failures that no longer reflect the current optimization state of the policy, making resampling less targeted.

The best performance is achieved at a moderate window size, indicating that effective resampling should balance candidate diversity and recency. Overall, these results show that ReCon is robust to the choice of W and benefits from retaining failed samples from a reasonably sized range of recent batches.

G.2 Sensitivity to the Uncertainty-Closeness Weight

We next analyze the effect of the weighting factor α_e, which balances the entropy-based uncertainty signal and the preference-based closeness signal in trajectory selection. Larger α_e assigns more weight to uncertainty, whereas smaller α_e places more emphasis on closeness.

As shown in Table 4, ReCon achieves the best performance at a moderate value of $\alpha_e = 0.4$. This result suggests that effective trajectory selection should remain primarily guided by closeness, while still incorporating a moderate amount of uncertainty information. When α_e is too small, the model relies almost entirely on closeness, which is already a strong signal but does not fully exploit potentially useful uncertainty cues. In contrast, as α_e increases, the method places growing emphasis on uncertainty, and performance drops noticeably when uncertainty becomes dominant.

This trend indicates that closeness is the more reliable signal for identifying correctable failed trajectories, while uncertainty is better viewed as a complementary cue rather than the primary selection criterion. Overall, the results support a closeness-dominant but balanced design, where a moderate contribution from uncertainty yields the strongest performance.

G.3 Sensitivity to the Trigger Threshold

We further analyze the sensitivity of ReCon to the trigger threshold ϵ_{stop}, which determines how pronounced the type-wise imbalance must be before resampling is activated. A larger ϵ_{stop} makes resampling more conservative, whereas a smaller value activates it more readily. We use $\epsilon_{\text{stop}} = 5 \times 10^{-3}$ in all main experiments.

As shown in Table 5, ReCon remains stable across the tested range. The F1 score varies by at most 0.86 points, and the resampling trigger rate varies by only 1.93 percentage points. The default value $\epsilon_{\text{stop}} = 5 \times 10^{-3}$ achieves the best performance. As expected, increasing the threshold slightly reduces the trigger rate, since resampling is activated only when the imbalance between conflict types becomes more pronounced.

Overall, these results show that ReCon does not require careful tuning of ϵ_{stop}. Moderate changes to the threshold only marginally affect how often resampling is triggered and lead to small differences in final performance.

Table 6: Multi-seed results on ConflictBANK. We report the mean and sample standard deviation over three random seeds (42, 1024, and 65535). Best results for each backbone are in bold.

Backbone	Method	IC	CM	ALL
Qwen2.5-7B-Instruct	KCR	82.44 ± 0.80	73.56 ± 1.20	78.54 ± 1.03
	ReCon	**82.64** ± 1.15	**82.70** ± 0.78	**82.60** ± 0.68
LLaMA-3.1-8B-Instruct	Knowledgeable-R1	67.10 ± 1.11	**77.12** ± 1.04	72.22 ± 0.86
	ReCon	**79.79** ± 0.87	76.91 ± 0.93	**77.53** ± 0.53

Table 7: Compute-matched comparison on ConflictBANK using Qwen2.5-7B-Instruct. For a batch of B prompts, all methods use at most $8B$ rollouts per training step.

Method	Rollout budget	IC	CM	ALL
GRPO	$8B$ full	78.43	74.15	76.59
KCR	$8B$ full	**82.20**	74.60	79.33
Knowledgeable-R1	$4B$ PK + $4B$ CK	73.21	76.76	74.95
ReCon	$5B$ initial + $\leq 3B$ branched	81.76	**82.48**	**82.43**

H Multi-Seed Stability

To assess whether the improvements of ReCon are stable across random seeds, we rerun the two closest type-level comparisons on ConflictBANK with three random seeds (42, 1024, and 65535): KCR versus ReCon with Qwen2.5-7B-Instruct, and Knowledgeable-R1 versus ReCon with LLaMA-3.1-8B-Instruct. KCR and Knowledgeable-R1 specialize in IC and CM conflicts, respectively, and are the strongest baselines on their favored conflict types. In the main results, these specialized baselines show only marginal advantages on their favored types, whereas the type-level differences in the other comparisons are substantially larger. These two settings are therefore the most informative for testing whether the observed type-level differences are reliable. Table 6 reports the mean and sample standard deviation over the three seeds. Since the main results in Table 1 are obtained from a single run, the mean values here differ slightly from those reported in the main text. On their favored conflict types, the specialized baselines and ReCon perform comparably, with differences well within one standard deviation. On the other conflict type, however, ReCon outperforms them by a large margin that far exceeds the seed-level variation, leading to higher overall scores on both backbones. Overall, these results indicate that the advantage of ReCon is not an artifact of a single favorable run.

I Compute-Matched Comparison

A natural concern is whether the gains of ReCon simply come from the additional rollouts it allocates. To examine this, we compare all methods under a unified budget of at most eight rollouts per prompt. For a batch of B prompts, GRPO and KCR generate $8B$ rollouts per step, so their results in the main experiments are directly comparable. ReCon generates $5B$ initial rollouts and at most $3B$ branched rollouts, so its total budget does not exceed $8B$. Knowledgeable-R1 originally samples 32 trajectories per example; we rerun it with four PK and four CK trajectories while keeping its original objectives and optimization settings. Since ReCon uses a different rollout configuration from the main experiments, its results here differ slightly from those in Table 1. As shown in Table 7, ReCon achieves the best CM and overall scores under the same maximum budget, while KCR remains slightly better on IC. This indicates that the effectiveness of ReCon cannot be attributed simply to using more rollouts.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's main contributions which are supported by the method described in Section 3 and the experimental results in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in Appendix **??**.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not contain theoretical results that require formal proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide complete method details in Section 3, training and evaluation setup in Section 4, dataset construction in Appendix A, evaluation metrics in Appendix B, and prompt templates in Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Code is not released at submission time to preserve anonymity, but will be made publicly available upon acceptance. The paper provides full method details, hyperparameters, and prompt templates sufficient for independent reimplementation.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: All training and evaluation details, including backbones, batch size, learning rate, rollout configuration, and decoding parameters, are provided in Section 4 and the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We report results from a single training run for each configuration due to the substantial computational cost of RLVR training across multiple backbones and benchmarks. The relative gains are consistent across three backbones (Qwen2.5-3B, Qwen2.5-7B, Llama3.1-8B) and four benchmarks, mitigating the risk of seed-specific artifacts.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the hardware (8× NVIDIA A800 GPUs) and training configuration in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research uses publicly available benchmarks and conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work focuses on improving reinforcement learning optimization for knowledge-conflict scenarios, and we do not foresee direct societal impacts beyond those of general LLM and RLVR research.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release any new pre-trained models, datasets, or other artifacts with high misuse risk. All experiments use publicly available models and benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly credit the creators and original owners of the datasets, models, and codebases used in this work by citing their original papers in the reference list.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not release new datasets, code, or model assets as part of the submission.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or research with human subjects, and therefore no IRB approval or equivalent review was required.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs are a core component of this work. We describe their usage in both method development and dataset construction, including memory-state probing, counterfactual context construction, and RLVR-based policy optimization on backbone LLMs.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.