Beyond Correctness: A Framework for Analyzing Reasoning and Faithfulness of Multi-hop Question Answering

Anonymous ACL submission

Abstract

Diagnosing the root causes of failures in Retrieval-Augmented Generation (RAG) is challenging, as current evaluations often conflate logical reasoning errors with the failure to adhere to retrieved context. To address this, we propose **ReFa**, a framework inspired by Structural Causal Models to explicitly disentangle *Reasoning* from *Faithfulness*. Through this lens, we characterize two distinct failure modes: the **'Fool'**, representing a deficiency in logical reasoning, and the **'Lazy'**, representing a lapse in faithfulness where the model defaults to parametric priors. To operationalize this diagnosis, we introduce Dual Reasoning Chain Editing, a mechanism that constructs controlled proxy chains to isolate reasoning structure from evidence faithfulness. We apply this method on **ReFaBench**, a novel benchmark constructed in this work featuring factual, counterfactual, and knowledge-conflicting scenarios. Furthermore, we propose **ReFa-DPO**, a decoupled preference optimization strategy that leverages these proxies to target specific failure patterns. Experimental results demonstrate that ReFa-DPO enhances robustness, particularly in mitigating parametric interference, by simultaneously enhancing both contextual faithfulness and reasoning capabilities.

1 Introduction

In recent years, Large Language Models (LLMs) have demonstrated strong performance across a wide range of domains (Zhao et al., 2023; Wu et al., 2024). In particular, RAG (Guu et al., 2020) enables LLMs to generate responses grounded in externally provided context, with the critical goal of prioritizing retrieved evidence over parametric memory (Zhang et al., 2025b) to mitigate hallucinations (Huang et al., 2025).

However, RAG systems face a fundamental tension when retrieved context conflicts with parametric knowledge or requires non-trivial synthesis across multiple pieces of information. Diagnosing the root causes of failures in such scenarios is challenging. As illustrated in Figure 1, we conceptualize two distinct failure modes: the **'Lazy'** mode, representing a **faithfulness failure** where the model ignores retrieved evidence to succumb to outdated parametric priors (e.g., *Camp Nou*); and the **'Fool'** mode, representing a **reasoning failure** where the model faithfully attends to the context but fails to resolve logical constraints (e.g., *former* vs. *now*), thus misidentifying the answer.

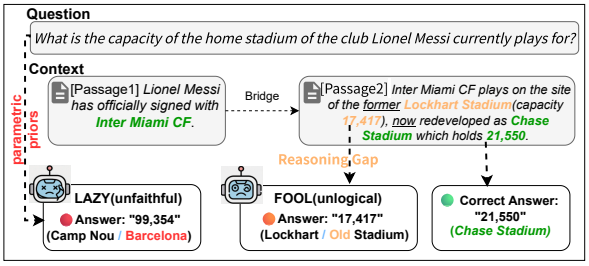

Figure 1: Illustration of the 'Lazy' and 'Fool' failure modes in a multi-hop RAG scenario.

Current evaluation paradigms are insufficient to disentangle these distinct error mechanisms. On one hand, classic Multi-hop Question Answering (MHQA) benchmarks such as HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020) and MuSiQue (Trivedi et al., 2022) conflate failures stemming from unlogical reasoning structure with those arising from unfaithful reliance on contextual evidence. On the other hand, robustness-oriented benchmarks including CofCA (Wu et al., 2025a) and ConFiQA (Bi et al., 2025) introduce counterfactual settings but treat model failure in a coarse-grained method. Despite recent efforts to evaluate intermediate steps (Liu et al., 2025b) or decouple memory from reasoning using special tokens (Jin et al., 2025), these methods still largely emphasize overall performance correctness, thereby lacking the granularity required to separately attribute fail-

ures to contextual faithfulness or logical reasoning. Without such diagnostic separation, it remains unclear whether downstream improvements should primarily target logical reasoning or faithfulness to contextual evidence.

To bridge this gap, we introduce **ReFa** (**Re**asoning and **Fa**ithfulness), a diagnostic framework for analyzing failure modes in multi-hop question answering. Inspired by Structural Causal Models, ReFa provides a structured abstraction that facilitates separate analysis of reasoning-related and faithfulness-related behaviors. We make our code and benchmark publicly available.[1]

In summary, our contributions are three-fold:

(1) We propose **ReFaBench**, an automated MHQA benchmark. Beyond standard factual settings, it systematically incorporates **counterfactual** and **knowledge-conflicting** scenarios, enabling controlled evaluation of model behavior under strong parametric interference.

(2) We introduce a **Dual Reasoning Chain Editing** method. By constructing *reasoning-centric* and *faithfulness-centric* inference paths as *diagnostic proxies*, this method allows us to isolate reasoning structure errors from faithfulness lapses, providing a fine-grained attribution of failure.

(3) We propose **ReFa-DPO**, a targeted preference optimization strategy that leverages these diagnostic signals. By aligning models with decoupled preference feedback targeting specific failure modes, ReFa-DPO consistently improves robustness across challenging evaluation scenarios.

2 Related Work

2.1 Reasoning and Faithfulness in LLMs

While Chain-of-Thought (CoT) prompting (Wei et al., 2022) improves multi-step reasoning performance (Zhang et al., 2023), generated reasoning chains often exhibit unfaithfulness, manifesting as post-hoc rationalizations rather than faithful use of supporting evidence (Arcuşchin et al., 2025; Xiong et al., 2025; Zhang et al., 2025b). This issue becomes particularly pronounced under knowledge conflicts (Wu et al., 2025b,a), where strong parametric priors contradict external contextual evidence. To better understand reasoning behavior, prior work has sought to disentangle different aspects of model competence, such as separating memory retrieval from reasoning ability (Jin et al., 2025), or to quantify reasoning faithfulness at the behavioral level (Schimanski et al., 2024; Lanham et al., 2023), as well as improve alignment through preference-based optimization (Rafailov et al., 2023; Bi et al., 2025).

However, existing approaches treat model failure as a coarse-grained signal, failing to distinguish logical reasoning deficits from faithfulness lapses. This lack of diagnostic granularity motivates our framework to explicitly disentangle these underlying error sources.

2.2 MHQA and Counterfactual Benchmarks

MHQA benchmarks such as HotpotQA (Yang et al., 2018) and MuSiQue (Trivedi et al., 2022) necessitate the integration of dispersed evidence to answer complex queries. Accordingly, evaluation extends beyond final answer to examine intermediate reasoning processes (Liu et al., 2025b; Wu et al., 2025a). Meanwhile, to mitigate data contamination and reliance on parametric memory, recent works have introduced counterfactual benchmarks like CofCA (Wu et al., 2025a) or knowledge-conflicting datasets such as ConFiQA (Bi et al., 2025).

However, prior research typically focuses on either counterfactual or conflicting scenarios in isolation. More importantly, these benchmarks often lack the diagnostic granularity to differentiate between errors in logical reasoning and lapses in contextual faithfulness, even when intermediate steps are evaluated. They often fail to determine whether a failure stems from a breakdown in logical reasoning ('Fool') or an unfaithful rejection of the premise ('Lazy'). Our ReFaBench fills this gap by providing a unified testbed with diagnostic proxies to disentangle these underlying deficits.

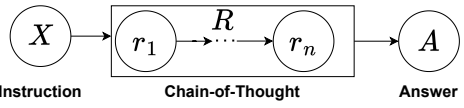

Figure 2: SCM that illustrates the interplay of reasoning ability and contextual faithfulness in LLMs.

3 Preliminaries: Structural Causal Model in LLMs

Structural Causal Models (SCMs) (Pearl et al., 2016) offer a general framework for describing dependencies among variables. In this work, we adopt an *SCM-inspired abstraction* to support the analysis of MHQA, focusing on two complementary aspects of the generation process: reasoning ability and contextual faithfulness.

[1] https://anonymous.4open.science/r/ReFa-3900/

As shown in Figure 2, X denotes the input instruction, including the context and prompt, while A denotes the answer generated by the LLM. Following prior work (Zhang et al., 2025a; Bao et al., 2025), we treat the CoT produced by the model as an intermediate variable R between X and A, where $R = (r_1, \ldots, r_n)$ and r_i represents the i-th reasoning step.

To diagnostically analyze the generation process, we **conceptually decompose** the probability of generating a reasoning chain. While a standard language model computes $P(r_i \mid r_{<i}, X)$ jointly, we approximate this dependency by factorizing it into two disentangled components (see Appendix A for the detailed derivation):

$$P(A \mid X) \propto \prod_{i=1}^{n} \underbrace{P(r_i \mid r_{<i})}_{\text{RA}} \cdot \underbrace{P(r_i \mid X)}_{\text{CF}}. \quad (1)$$

This decomposition empirically highlights two dominant and partially separable capabilities: **Reasoning Ability (RA)**, denoted by $P(r_i \mid r_{<i})$, quantifies the internal capacity to construct valid logical chains ($\rightarrow$ 'Fool'); while **Contextual Faithfulness (CF)**, denoted by $P(r_i \mid X)$, represents the adherence to retrieved evidence over parametric priors ($\rightarrow$ 'Lazy'). These terms guide the design of our diagnostic proxies (R_R and R_F) in Section 4.4.

4 The ReFa Framework

We propose **ReFa**, a unified framework designed to diagnose and mitigate failures in reasoning ability and contextual faithfulness. As illustrated in Figure 3, the framework comprises two synergistic pillars: **(1) ReFaBench** serves as a comprehensive diagnostic testbed, integrating an automated *Data Construction Pipeline* to generate multi-hop question answers across factual, counterfactual, and knowledge-conflicting scenarios, alongside a *Dual Reasoning Chain Editing* module that constructs diagnostic proxies (R_R and R_F) to explicitly disentangle 'Fool' and 'Lazy' failure modes. **(2) ReFa-DPO** acts as the corresponding mitigation strategy, which leverages these diagnostic proxies for *Preference Pair Construction* and performs *Decoupled Optimization* to simultaneously bolster reasoning ability and contextual faithfulness, thereby enhancing overall model robustness.

4.1 Data Construction Pipeline

Real-World Fact Sampling We construct our fact corpus based on the Wikipedia data source specified by CofCA (Wu et al., 2025a). Documents are selected and filtered to ensure sufficient factual richness and suitability for MHQA, with an emphasis on high-visibility content to support realistic knowledge-conflicting scenarios. Detailed sampling criteria and filtering procedures are provided in Appendix B.

Data Generation, Polishing and Validation Inspired by prior work on multi-hop question answering (Wu et al., 2025a; Liu et al., 2025b; Shen et al., 2025), we adopt an automated prompt-based pipeline of data generation and polishing. For each sampled document D, we apply both counterfactual and conflicting-factual transformations, resulting in three paired (**D**ocument, **Q**uestion, **A**nswer) datasets:

- $(D_{\text{fact}}, Q_{\text{fact}}, A_{\text{fact}})$: We generate 2-, 3-, and 4-hop bridge and comparison questions based on the original factual documents.
- $(D_{\text{cnt}}, Q_{\text{cnt}}, A_{\text{cnt}})$: Following CofCA (Wu et al., 2025a), we use entities and attributes from D_{fact} as editing anchors to construct plausible but non-existent alternatives.
- $(D_{\text{conf}}, Q_{\text{conf}}, A_{\text{conf}})$: A dynamic counterfactual injection agent identifies key factual elements from $(D_{\text{fact}}, Q_{\text{fact}}, A_{\text{fact}})$ and deliberately introduces knowledge conflicts that contradict the model's parametric world knowledge (Ming et al., 2025).

This pipeline results in a benchmark that evaluates model performance not only under standard factual conditions, but also in settings where parametric knowledge conflicts with the provided context. By incorporating diverse counterfactual and conflicting scenarios, the benchmark enables simultaneous evaluation under conflicting and out-of-distribution (OOD) conditions. To ensure data quality, all generated instances undergo automated polishing and a human validation for quality assessment, with complete details provided in Appendices C and D.

4.2 Dual Reasoning Chain Editing

To disentangle reasoning errors from failures of contextual faithfulness in MHQA, we construct controlled variants of CoT that serve as diagnostic reference proxies.

We build a reference CoT using teacher LLMs, denoted as R_{golden}. It is synthesized from verified

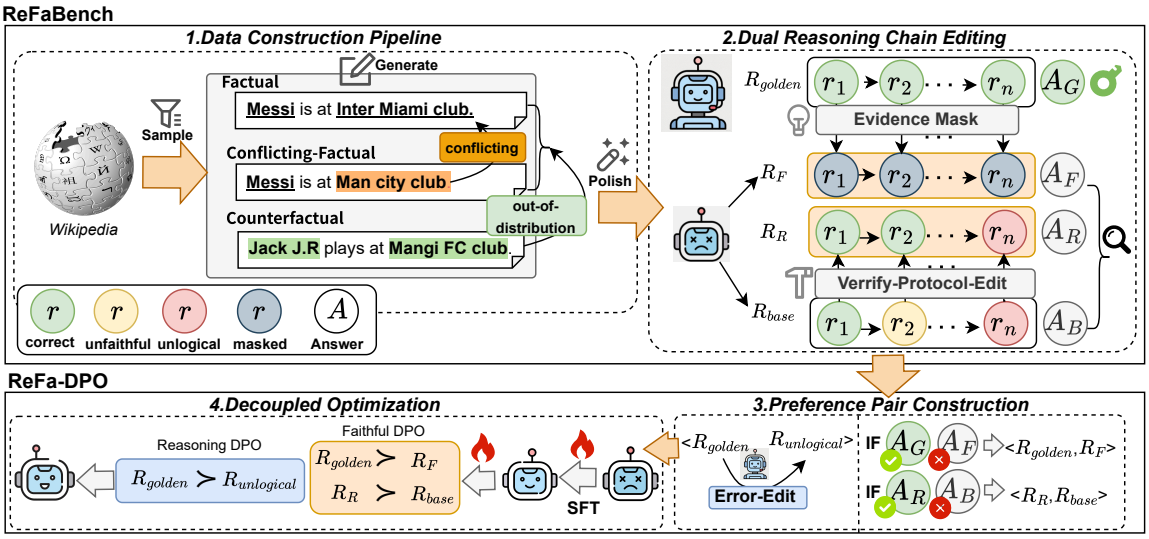

Figure 3: The main components and pipelines of our ReFa framework.

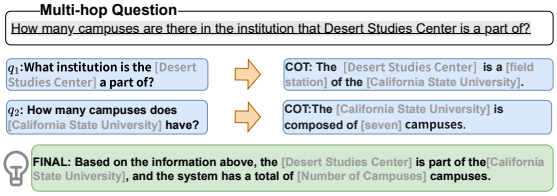

Figure 4: Evidence masking in the Faithfulness-centric R_{F}, where [Gray] denotes the masked content. It preserves the reasoning structure and evaluates faithfulness.

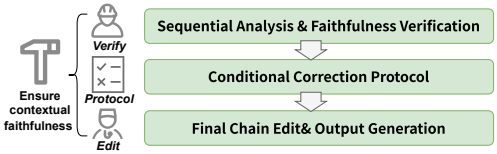

Figure 5: Verify–Protocol–Edit workflow for constructing the Reasoning-centric R_{R}. It ensures contextual faithfulness and evaluates reasoning ability.

sub-QA pairs and refined into atomic reasoning steps r_i aligned with each sub-question q. The resulting chain $R_{\text{golden}} = (r_1, \ldots, r_n, A)$ provides a validated reasoning trajectory ending with the gold answer A. In contrast, we generate a baseline CoT R_{base} directly from the evaluated LLM, reflecting the model's native, unconstrained behavior which may contain entangled errors.

Based on these chains, we derive two controlled variants CoTs via targeted interventions:

R_{F}: Faithfulness-centric. Starting from the golden chain R_{golden}, we selectively mask evidence-bearing spans to construct a reasoning template, as illustrated in Figure 4. The model is tasked with reconstructing the masked content using the provided documents, while the *logical structure* is fixed by the template. With the logical structure fixed, R_{F} isolates the model's capacity for evidence retrieval and usage. Thus, it serves as a proxy for **Contextual Faithfulness (CF)**, exposing the 'Lazy' mode.

R_{R}: Reasoning-centric. Starting from the student-generated R_{base}, we apply a *Verify–Protocol–Edit* workflow (Figure 5) to correct steps that conflict with the provided context. This enforces contextual faithfulness while preserving the model's native multi-step reasoning trajectory. By minimizing unfaithful evidence usage, R_{R} serves as a proxy for **Reasoning Ability (RA)**, exposing the 'Fool' mode.

Together, these proxies enable fine-grained failure attribution: R_{F} isolates faithfulness from reasoning, while R_{R} does the inverse. See Appendix J for case studies, and the prompts of the dual reasoning chain editing are detailed in Appendix L.

Benchmark	Data Scenarios			Diagnostic Capabilities		
	Fact	Conf	Cnt	Corr	RA	CF
HotpotQA	✓	–	–	✓	–	–
2WikiMultihopQA	✓	–	–	✓	–	–
MuSiQue	✓	–	–	✓	–	–
ConFiQA	–	✓	–	✓	–	–
CofCA	✓	–	✓	✓	–	–
ReFaBench (Ours)	✓	✓	✓	✓	✓	✓

Note: **Fact**: Factual, **Conf**: Conflicting-Factual, **Cnt**: Counterfactual. **Corr**: Correctness, **RA**: Reasoning Ability, **CF**: Contextual Faithfulness. '–' indicates the feature is absent.

Table 1: Comparison of data scenarios and diagnostic capabilities between ReFaBench and existing MHQA benchmarks.

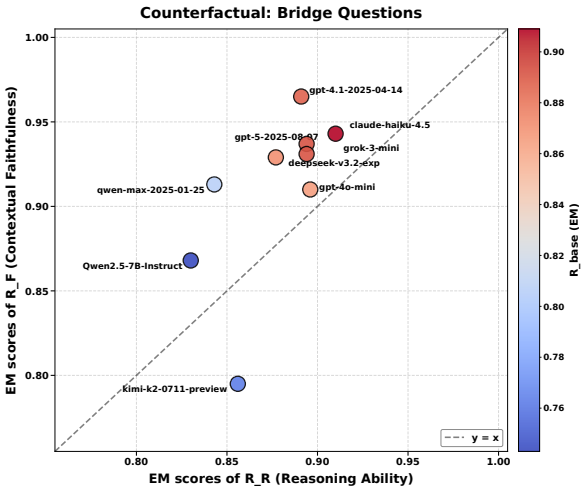

Figure 6: Performance comparison on counterfactual Bridge questions.

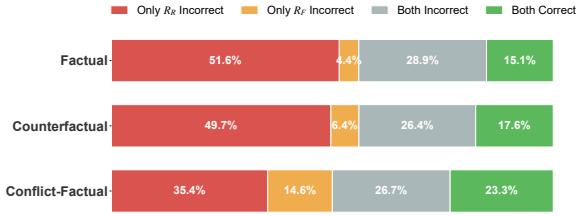

Figure 7: Attributing Failures via Dual Proxies on Qwen2.5-7B-Instruct.

4.3 Overview of ReFaBench

Table 1 presents a comparative analysis of ReFaBench against representative MHQA benchmarks, highlighting differences in data scenarios and diagnostic capabilities.

Data Scenarios Unlike prior datasets focused on in-domain facts, ReFaBench incorporates two challenging settings to probe distinct failure modes: (1) *Conflicting-Factual*, where the context explicitly contradicts parametric priors to maximize interference; and (2) *Counterfactual*, depicting plausible but non-existent situations outside the pretraining distribution. Notably, these out-of-distribution scenarios play a pivotal role in evaluating robustness, as detailed in Section 5.

Diagnostic Capabilities Moving beyond simple answer correctness, ReFaBench supports disentangled diagnosis of reasoning and faithfulness via dedicated proxies. For answer-level evaluation, we adopt Exact Match (EM) as the primary metric of comparing model diagnostic capabilities (Rajpurkar et al., 2016), with complementary F1 scores reported in Appendix E.

4.4 Evaluation on ReFaBench

We evaluate ReFaBench across a diverse set of state-of-the-art LLMs, encompassing both proprietary and open-source models[2]. Our analysis transcends binary answer correctness to examine model behavior along two complementary dimensions: reasoning ability and contextual faithfulness.

Reasoning–Faithfulness Quadrant Analysis Figure 6 projects each model into a two-dimensional space defined by reasoning-centric (R_R) and faithfulness-centric (R_F) performance. We observe an asymmetric progression between the two dimensions. While better models generally perform higher on both, R_F (y-axis) consistently outpaces R_R (x-axis) across most models, identifying multi-hop reasoning as the primary bottleneck. The color intensity represents the overall answer correctness (R_{base}), showing that strong end-task performance requires balanced capability along both axes, which motivates the necessity of dual-axis analysis. See Appendix F for an extended analysis.

Attributing Failures via Dual Proxies Figure 7 characterizes the error distribution for Qwen2.5-7B-Instruct through our diagnostic lens. Notably, 84.9% of all incorrect answers are explicitly captured by at least one diagnostic proxy (R_{R} or R_{F}). This high diagnostic coverage demonstrates that the proposed dual-proxy design effectively encapsulates the primary drivers of MHQA failures, specifically the 'Fool' and 'Lazy' modes. These results underscore ReFa's capacity to provide granular visibility into model behavior that is typically obscured by conventional correctness metrics.

Robustness under Parametric Interference Table 2 reports model performance across factual, counterfactual, and conflicting-factual settings (see full results in Appendix E). By manipulating the alignment between external context and internal priors, we observe distinct degradation patterns. Most models exhibit significant performance drops under conflicting-factual conditions compared to counterfactual ones, revealing that *parametric priors* (the '**Lazy**' failure) poses a greater challenge

[2]The evaluated models include GPT-4.1-2025-04-14 (OpenAI, 2025), GPT-5-2025-08-07 (OpenAI, 2025), GPT-4o-mini (OpenAI, 2024), Claude-haiku-4.5 (Anthropic, 2025), Qwen2.5-7B-Instruct (Team, 2025), Qwen-max-2025-01-25 (Cloud, 2025), Grok-3-mini (xAI, 2025), kimi-k2-0711-preview (Team et al., 2025), and DeepSeek-V3.2-exp (DeepSeek-AI, 2025).

than processing novel information. Detailed key findings are summarized in Section 4.5.

Model	Type	Metric	Factual (Score)	Relative Drop (Δ%)	
				Cnt	Conf
claude-haiku-4-5	Bridge	R_{base}	0.937	-3.0%	**-3.1%**
		R_R	0.939	-3.1%	-0.2%
		R_F	0.966	-2.4%	**-1.3%**
	Comparison	R_{base}	0.814	-3.1%	-11.2%
		R_R	0.815	-2.9%	-7.0%
		R_F	0.861	0.0%	-4.2%
gpt-5-2025-08-07	Bridge	R_{base}	**0.943**	-5.4%	-11.7%
		R_R	**0.949**	-5.8%	-1.4%
		R_F	**0.970**	-3.4%	-1.6%
	Comparison	R_{base}	0.788	-1.3%	**-6.1%**
		R_R	0.789	-1.4%	**-2.9%**
		R_F	0.908	+0.4%	**-1.1%**
deepseek-v3.2-exp	Bridge	R_{base}	0.917	-5.0%	-5.9%
		R_R	0.920	-4.7%	-0.4%
		R_F	0.961	-3.3%	-1.8%
	Comparison	R_{base}	**0.815**	**+0.9%**	-12.3%
		R_R	**0.819**	+0.5%	-7.2%
		R_F	**0.913**	+0.5%	-11.0%
Qwen2.5-7B-Instruct	Bridge	R_{base}	0.758	**-2.0%**	-6.6%
		R_R	0.827	+0.4%	+2.4%
		R_F	0.876	**-0.9%**	-2.3%
	Comparison	R_{base}	0.607	-8.1%	-26.5%
		R_R	0.661	-3.8%	-8.5%
		R_F	0.805	**+1.0%**	-11.6%

Table 2: Performance comparison of representative LLMs. Significant performance drops ($> 10\%$) are highlighted in red. The **best** results (highest factual score or best robustness) are marked in bold.

4.5 Key Findings

Our analysis highlights several consistent patterns in model behavior on ReFaBench, shedding light on the mechanisms of *parametric interference* (conflicts between pre-trained priors and external context), *parametric inertia* (difficulty in overriding entrenched knowledge), and *reasoning deficits* (limitations in constructing valid logical chains).

- **Finding 1: Parametric interference exacerbates faithfulness deficits under conflicting scenarios.** Compared to counterfactual (Cnt) scenarios, conflicting-factual (Conf) settings generally induce significantly larger performance degradation. This suggests that direct contradictions with **parametric priors** pose a greater challenge than distributional novelty alone, often causing models to **succumb to internal memory** rather than adhering to the provided evidence. Such behavior aligns with the '**Lazy**' failure mode, where the model fails to suppress internal knowledge despite retrieving correct context.

- **Finding 2: Reasoning structure and faithfulness degrade divergently.** A decline in final answer correctness (R_{base}) does not imply a parallel collapse in reasoning capability. On Bridge questions, `Qwen2.5` maintains or even slightly improves its reasoning consistency (R_R, $+2.4\%$) under knowledge conflict, while its base correctness drops significantly (-6.6%). This decoupling indicates that errors primarily stem from a lapse in **contextual adherence** (the '**Lazy**' mode) rather than an inability to construct valid logical chains.

- **Finding 3: Strong parametric priors do not guarantee robustness.** Superior in-domain performance does not translate to robustness against interference. Even state-of-the-art models such as `gpt-5` experience notable performance drops (e.g., -11.7%) when confronted with contradictory context. This suggests a trade-off associated with stronger parametric memory: higher **inertia** makes it more difficult for the model to override **entrenched priors** in favor of external evidence (Bi et al., 2025; Ming et al., 2025).

- **Finding 4: Structural guidance mitigates reasoning bottlenecks in weaker models.** For smaller models, a substantial portion of performance loss can be attributed to deficits in **logical chain construction** rather than evidence grounding. When provided with Ground-truth reasoning structures, `Qwen2.5` improves from $R_{\text{base}} = 0.607$ to $R_F = 0.805$ ($+33\%$), indicating that once the reasoning trajectory is fixed, the model can faithfully retrieve and integrate contextual information. This confirms that the dominant bottleneck in such cases is the '**Fool**' failure mode (unlogical reasoning).

5 Enhancing Robustness via ReFa-DPO

Moving beyond diagnostic evaluation, we investigate whether the disentangled failure signals identified by ReFaBench can be leveraged to mitigate **parametric interference**. Specifically, we explore whether aligning models to explicitly penalize the '**Lazy**' and '**Fool**' modes leads to more robust generalization compared to standard fine-tuning. All experiments are conducted on `Qwen2.5-7B-Instruct`, with implementation details provided in Appendix G.

5.1 Supervised Fine-Tuning: Benefits and Limitations

Table 3 shows that SFT substantially enhances end-task performance (R_{base}), with relative gains reach-

ing $+71.5\%$ in D_{conf} scenarios. However, these improvements are highly sensitive to the distribution similarity between training and evaluation datasets. Maximal gains typically occur when training and evaluation datasets align, which suggests a susceptibility to distributional specialization rather than the acquisition of robust reasoning capabilities.

To assess generalization, we evaluate models under counterfactual scenarios (D_{cnt}) as an OOD proxy. A task-dependent pattern emerges: Bridge questions benefit more from factual supervision to stabilize logical construction, while Comparison tasks favor D_{conf} to mitigate parametric inertia.

Ultimately, while SFT improves in-distribution correctness, it treats MHQA as a coarse-grained process. Consequently, it lacks the necessary granularity to independently optimize reasoning ability and contextual faithfulness, a limitation that motivates the decoupled approach of ReFa-DPO.

Model	Type	Variant	R_{base}	R_R	R_F
Instruct	Bridge	Factual	0.758	0.827	0.876
		Cnt	0.743	0.830	0.868
		Conf	0.708	0.847	0.856
	Comp.	Factual	0.607	0.661	0.805
		Cnt	0.558	0.636	0.813
		Conf	0.446	0.605	0.712
SFT	Bridge	Factual	**0.842 (+11.1%)**	**0.895 (+8.2%)**	**0.904 (+3.2%)**
		Cnt	**0.795 (+7.0%)**	**0.865 (+4.2%)**	**0.881 (+1.5%)**
		Conf	0.839 (+18.5%)	0.902 (+6.5%)	0.911 (+6.4%)
	Comp.	Factual	**0.784 (+29.1%)**	**0.808 (+22.2%)**	**0.881 (+9.4%)**
		Cnt	0.751 (+34.6%)	**0.793 (+24.7%)**	0.869 (+6.9%)
		Conf	0.687 (+54.0%)	0.715 (+18.2%)	0.811 (+13.9%)
SFT-Conf	Bridge	Factual	0.818 (+7.9%)	0.891 (+7.7%)	0.902 (+3.0%)
		Cnt	0.774 (+4.2%)	0.858 (+3.4%)	0.871 (+0.3%)
		Conf	**0.874 (+23.5%)**	**0.918 (+8.4%)**	**0.944 (+10.3%)**
	Comp.	Factual	0.782 (+28.8%)	0.808 (+22.2%)	0.871 (+8.2%)
		Cnt	**0.757 (+35.7%)**	0.785 (+23.4%)	**0.876 (+7.7%)**
		Conf	**0.765 (+71.5%)**	**0.757 (+25.1%)**	**0.835 (+17.3%)**

Table 3: Performance of SFT variants. Models are based on `Qwen2.5-7B` and abbreviated as **Instruct**, **SFT**, and **SFT-Conf**. Values are EM scores, with relative gains over Instruct in parentheses.

5.2 ReFa-DPO: Disentangled Preference Optimization

5.2.1 Limitations of Standard DPO

While SFT improves overall correctness, standard DPO yields only marginal gains (Table 4) due to *failure-mode entanglement* (Feng et al., 2024; Yuan et al., 2024): it treats all errors as a coarse-grained negative signal, conflating deficits in *reasoning structure* (**Fool**) with unfaithful reliance on *parametric priors* (**Lazy**), which leads to ambiguous preference gradients and hinders targeted correction of reasoning or faithfulness.

5.2.2 Preference Pair Construction

To address the aforementioned limitation, we propose **ReFa-DPO**, a strategy that constructs two distinct preference datasets to decouple failure modes.

Reasoning-Centric Preferences ($\mathcal{D}_{\text{reason}}$) We construct $\mathcal{D}_{\text{reason}}$ by pairing the golden chain R_{golden} (y_w) with a synthesized unlogical chain $R_{\text{unlogical}}$ (y_l) to address logical deficiencies. As detailed in Appendix L, $R_{\text{unlogical}}$ acts as a hard negative containing plausible but invalid logical steps. This forces the model to distinguish rigorous reasoning from superficial fluency or contextual faithfulness, effectively targeting the 'Fool' failure mode.

Faithfulness-Centric Preferences ($\mathcal{D}_{\text{faith}}$) To penalize parametric inertia, we construct $\mathcal{D}_{\text{faith}}$ by contrasting context-adhering responses (y_w) with unfaithful ones driven by internal priors (y_l). This set includes (1) $\langle R_{\text{golden}}, R_{\text{F}} \rangle$ pairs where the model fails the faithfulness proxy, and (2) $\langle R_{\text{R}}, R_{\text{base}} \rangle$ pairs where the model succumbs to priors despite possessing sufficient reasoning capacity (R_{R} correct). This strategy targets the 'Lazy' mode by prioritizing evidence-grounded trajectories over memory shortcuts.

5.2.3 Decoupled Optimization Objective

ReFa-DPO integrates these distinct supervision signals by optimizing the standard DPO objective over the joint preference set $\mathcal{D} = \mathcal{D}_{\text{reason}} \cup \mathcal{D}_{\text{faith}}$:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\log\sigma\Big(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\Big)\Big], \tag{2}$$

where β controls the deviation from the reference model π_{ref}. By optimizing this joint objective, the model learns to simultaneously bolster logical structure and contextual faithfulness without mutual interference.

5.3 Results and Findings

Table 4 presents a comparative analysis of DPO-based alignment methods. To ensure a fair evaluation, all models are initialized from the same SFT baseline and trained under a strictly controlled data budget of 1,200 preference samples. Overall, ReFa-DPO consistently outperforms both standard DPO (Rafailov et al., 2023) and ContextDPO (Bi et al., 2025) across all question types and knowledge conditions.

Model	Question Type	Factual			Counterfactual			Conflicting-Factual		
		R_{base}	R_R	R_F	R_{base}	R_R	R_F	R_{base}	R_R	R_F
`Qwen2.5-SFT`	Bridge	0.842	0.895	0.904	0.795	0.865	0.881	0.839	0.902	0.911
	Comparison	0.784	0.808	0.881	0.751	0.793	0.869	0.687	0.715	0.811
`Qwen2.5-SFT-DPO`	Bridge	0.840	0.903	0.902	0.799	0.869	**0.883**	0.825	0.899	0.911
	Comparison	0.775	0.812	0.873	0.739	0.782	0.873	0.689	0.735	0.805
`Qwen2.5-SFT-ContextDPO`	Bridge	0.836	0.896	0.894	0.798	0.871	0.881	0.844	0.910	**0.927**
	Comparison	0.781	0.812	0.882	0.740	0.780	0.881	0.711	0.740	0.848
`Qwen2.5-SFT-ReFa`	Bridge	**0.862**	**0.919**	0.910	**0.812**	**0.882**	0.881	**0.858**	**0.926**	0.920
	Comparison	**0.793**	**0.822**	**0.888**	**0.757**	**0.794**	**0.883**	**0.718**	**0.742**	**0.842**
Ablation of Components										
`w/o Re(<`R_{golden}`,`$R_{unlogical}$`>)`	Bridge	0.854	0.903	0.906	0.796	0.871	0.878	0.842	0.904	0.918
	Comparison	0.789	0.814	0.881	0.743	0.791	0.876	0.694	0.726	0.822
`w/o Fa(<`R_R`,`R_{base}`>)`	Bridge	0.861	0.909	**0.912**	0.809	0.877	0.881	0.853	0.922	0.916
	Comparison	0.789	0.819	0.883	0.752	0.792	0.880	0.702	0.745	0.823
`w/o Fa(<`R_{golden}`,`R_F`>)`	Bridge	0.856	0.916	0.910	0.802	0.877	0.882	0.848	0.915	0.919
	Comparison	0.788	0.819	0.884	0.752	0.794	0.875	0.697	0.740	0.822

Table 4: Synergy of SFT and DPO methods. Green background indicates improvement over SFT, while Red background indicates degradation. The bottom section presents the ablation study.

Our experiments yield two key findings:

- **Finding 1: ReFa-DPO achieves synergistic and robust improvements over SFT and coarse-grained alignment.** While SFT enhances MHQA correctness, especially in matched scenarios with gains up to $+51.6\%$ EM on conflicting-factual tasks, these improvements remain uneven and susceptible to performance regression under distributional shifts. Conventional coarse-grained optimization, such as standard DPO, offers marginal benefits and may even degrade reasoning or faithfulness due to signal entanglement. In contrast, ReFa-DPO consistently bolsters both reasoning ability (R_{R}) and contextual faithfulness (R_{F}) across all settings. It surpasses all other DPO variants and achieves state-of-the-art results among alignment methods, notably even without the need for domain-specific SFT pretraining. See Appendices H and I for further details.

- **Finding 2: ReFaBench enables a principled diagnostic-to-mitigation alignment pipeline.** ReFaBench utilizes dual-chain editing to construct targeted preference pairs that isolate reasoning deficits from faithfulness failures. By contrasting validated chains against unlogical ones (targeting 'Fool') and faithful outputs against unfaithful ones (targeting 'Lazy'), ReFa-DPO implements disentangled preference supervision. This design bridges diagnosis and mitigation by selectively penalizing distinct failure tendencies within a unified optimization framework.

5.4 Ablation Study

As shown in Table 4, removing either the reasoning-centric or faithfulness-centric component results in performance degradation. This confirms that both optimization objectives contribute synergistically to the robustness of ReFa-DPO, validating the necessity of the decoupled design.

6 Conclusion

In this work, we argue that failures in RAG are not coarse-grained but stem from two disentangled modes: deficiencies in **reasoning ability** ('**Fool**') and lapses in **contextual faithfulness** ('**Lazy**'). This distinction underscores that final answer correctness alone is insufficient for diagnosing reliability, particularly under knowledge conflicts.

To enable such fine-grained analysis, we introduce **ReFaBench**, a benchmark constructed in this work to disentangle reasoning from faithfulness across factual, counterfactual, and conflicting settings. Building on this, we propose **ReFa-DPO**, a strategy that explicitly decouples optimization for these modes, achieving superior robustness over coarse-grained DPO variants. Overall, **ReFa** provides a unified framework bridging diagnosis and mitigation to enhance RAG trustworthiness.

Limitations

(1) Currently, ReFaBench primarily focuses on factual multi-hop reasoning (Bridge and Comparison types) grounded in encyclopedic knowledge. We have not yet evaluated the framework's efficacy on other complex reasoning domains, such as mathematical derivation, commonsense reasoning, or code generation, where the definition of "faithfulness" and "parametric conflict" may require different formulations. Future work will aim to extend the Dual Reasoning Chain Editing framework to these broader domains. (2) Synthesizing high-quality QA pairs and CoT through teacher LLM APIs involves notable financial costs. As discussed in the **Computational Cost Analysis** (Appendix K), these expenses pose a challenge for large-scale dataset expansion. Future research should focus on developing cost-efficient generation strategies while maintaining the rigorous quality of the synthetic data.

Ethics Statement

Our benchmark includes *counterfactual* and *conflicting-factual* examples, which contain deliberately altered facts for diagnostic purposes only. These synthetic scenarios are designed solely to disentangle reasoning errors from faithfulness failures in RAG systems, and are not intended to propagate misinformation. All such examples are clearly identified as non-factual within our evaluation.

References

Anthropic. 2025. Claude 3.5 and later models.

Iván Arcuşchin, Jett Janiak, Robert Krzyzanowski, Senthooran Rajamanoharan, Neel Nanda, and Arthur Conmy. 2025. Chain-of-thought reasoning in the wild is not always faithful. *Preprint*, arXiv:2503.08679.

Guangsheng Bao, Hongbo Zhang, Cunxiang Wang, Linyi Yang, and Yue Zhang. 2025. How likely do llms with cot mimic human reasoning? In *COLING*, pages 7831–7850.

Baolong Bi, Shaohan Huang, Yiwei Wang, Tianchi Yang, Zihan Zhang, Haizhen Huang, Lingrui Mei, Junfeng Fang, Zehao Li, Furu Wei, Weiwei Deng, Feng Sun, Qi Zhang, and Shenghua Liu. 2025. Context-DPO: Aligning language models for context-faithfulness. In *Findings of ACL 2025*.

Alibaba Cloud. 2025. Qwen-max and qwen3-max series.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

DeepSeek-AI. 2025. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Duanyu Feng, Bowen Qin, Chen Huang, Zheng Zhang, and Wenqiang Lei. 2024. Towards analyzing and understanding the limitations of dpo: A theoretical perspective. *arXiv preprint arXiv:2404.04626*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *ICML*, pages 3929–3938.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *COLING*, pages 6609–6625.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu. Chen. 2022. Lora: Low-rank adaptation of large language models. In *ICLR*, page 3.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *TOIS*, 43:1–55.

Mingyu Jin, Weidi Luo, Sitao Cheng, Xinyi Wang, Wenyue Hua, Ruixiang Tang, William Yang Wang, and Yongfeng Zhang. 2025. Disentangling memory and reasoning ability in large language models. In *ACL*, pages 1681–1701.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamile Lukosiute, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, and 11 others. 2023. Measuring faithfulness in chain-of-thought reasoning. *Preprint*, arXiv:2307.13702.

Jiyuan Liu, Jielin Song, Yunhe Pang, Zhiyu Shen, and Yanghui Rao. 2025a. CARE: A disagreement detection framework with concept alignment and reasoning enhancement. In *EMNLP*, pages 13275–13290.

Qichuan Liu, Chentao Zhang, Chenfeng Zheng, Guosheng Hu, Xiaodong Li, and Zhihong Zhang. 2025b. Beyond the answer: Advancing multi-hop qa with fine-grained graph reasoning and evaluation. In *ACL*, pages 23433–23456.

Yifei Ming, Senthil Purushwalkam, Shrey Pandit, Zixuan Ke, Xuan-Phi Nguyen, Caiming Xiong, and Shafiq Joty. 2025. Faitheval: Can your language model stay faithful to context, even if" the moon is made of marshmallows". *ICLR*.

OpenAI. 2024. Gpt-4o system card.

OpenAI. 2025. Introducing gpt-5.2.

Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. 2016. *Causal inference in statistics: A primer*. John Wiley & Sons.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, pages 53728–53741.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Tobias Schimanski, Jingwei Ni, Mathias Kraus, Elliott Ash, and Markus Leippold. 2024. Towards faithful and robust LLM specialists for evidence-based question-answering. In *ACL*.

Zhiyu Shen, Jiyuan Liu, Yunhe Pang, and Yanghui Rao. 2025. Hopweaver: Synthesizing authentic multi-hop questions across text corpora. *arXiv preprint arXiv:2505.15087*.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, and 1 others. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

Qwen Team. 2025. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. musique: Multi-hop questions via single-hop question composition. *TACL*, 10:539–554.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, pages 24824–24837.

Jian Wu, Linyi Yang, Zhen Wang, Manabu Okumura, and Yue Zhang. 2025a. Cofca: A step-wise counterfactual multi-hop qa benchmark. In *ICLR*.

Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, and 1 others. 2024. A survey on large language models for recommendation. In *WWW*, page 60.

Xiaobao Wu, Liangming Pan, Yuxi Xie, Ruiwen Zhou, Shuai Zhao, Yubo Ma, Mingzhe Du, Rui Mao, Luu Anh Tuan, and William Yang Wang. 2025b. Antileakbench: Preventing data contamination by automatically constructing benchmarks with updated real-world knowledge. In *ACL*, pages 18403–18419.

xAI. 2025. Grok 3 beta — the age of reasoning agents.

Zidi Xiong, Shan Chen, Zhenting Qi, and Himabindu Lakkaraju. 2025. Measuring the faithfulness of thinking drafts in large reasoning models. *Preprint*, arXiv:2505.13774.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, pages 2369–2380.

Hui Yuan, Yifan Zeng, Yue Wu, Huazheng Wang, Mengdi Wang, and Liu Leqi. 2024. A common pitfall of margin-based language model alignment: Gradient entanglement. *arXiv preprint arXiv:2410.13828*.

Congzhi Zhang, Linhai Zhang, Jialong Wu, Yulan He, and Deyu Zhou. 2025a. Causal prompting: Debiasing large language model prompting based on front-door adjustment. In *AAAI*.

Qinggang Zhang, Zhishang Xiang, Yilin Xiao, Le Wang, Junhui Li, Xinrun Wang, and Jinsong Su. 2025b. Faithfulrag: Fact-level conflict modeling for context-faithful retrieval-augmented generation. In *ACL*, pages 21863–21882.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *ICLR*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, (2).

A Proof of Equation 1

We derive a factorized form of the answer probability $P(A \mid X)$ by explicitly modeling intermediate reasoning steps $R = (r_1, \ldots, r_n)$ in a Chain-of-Thought (CoT) process.

Step 1: Marginalization over reasoning chains By the law of total probability, the answer likelihood can be expanded over all possible reasoning chains:

$$\begin{aligned} P(A \mid X) &= \sum_R P(A, R \mid X) \\ &= \sum_R P(A \mid R, X)\, P(R \mid X). \end{aligned} \tag{3}$$

Step 2: Dominant-chain approximation In practice, autoregressive language models induce a distribution over reasoning chains that is often sharply peaked. We therefore adopt a dominant-chain approximation, replacing the summation by its highest-probability term R^*:

$$P(A \mid X) \approx P(A \mid R^*, X)\, P(R^* \mid X), \quad (4)$$

where $R^* = \arg\max_R P(R \mid X)$. This approximation is standard in analyses of structured latent-variable models.

Step 3: Autoregressive factorization of the reasoning chain To simplify notation for the remainder of the derivation, we analyze the probability of this dominant chain and drop the superscript, denoting R^* simply as R.

Using the chain rule, the probability of the reasoning chain factorizes as:

$$P(R \mid X) = \prod_{i=1}^{n} P(r_i \mid r_{<i}, X), \quad (5)$$

where $r_{<i} = (r_1, \ldots, r_{i-1})$ denotes the reasoning history up to step i.

Step 4: Decomposition into reasoning ability and contextual faithfulness We introduce a conceptual decomposition of each reasoning step into two complementary factors:

- **Reasoning ability** $P(r_i \mid r_{<i})$, capturing the model's intrinsic capacity to produce a logically coherent continuation given the prior reasoning history;
- **Contextual faithfulness** $P(r_i \mid X)$, capturing the compatibility of the reasoning step with the input context or evidence.

This decomposition is not claimed to reflect the true generative process of language models. Instead, it serves as a modeling abstraction that isolates logical coherence from contextual alignment, enabling analysis of distinct reasoning failure modes.

Assuming a product-of-experts (PoE) approximation, we write:

$$P(r_i \mid r_{<i}, X) \propto P(r_i \mid r_{<i}) \cdot P(r_i \mid X), \quad (6)$$

where the proportionality constant corresponds to a normalization term independent of r_i. Intuitively, this formulation enforces an *AND* constraint: a reasoning step is assigned high probability only if it is both logically coherent and faithful to the input.

Substituting this approximation into the chain factorization yields:

$$P(R \mid X) \propto \prod_{i=1}^{n} \left[P(r_i \mid r_{<i}) \cdot P(r_i \mid X)\right]. \quad (7)$$

Step 5: Near-determinism of the answer given reasoning Given the complete dominant reasoning chain R (i.e., R^*), the final answer A is typically explicitly contained in or deterministically inferred from the chain. We therefore assume:

$$P(A \mid R, X) \approx P(A \mid R) \approx 1, \quad (8)$$

reflecting that uncertainty in A is dominated by uncertainty in the reasoning process rather than in answer extraction.

Step 6: Final derivation Combining the above approximations (substituting Eq. (7) and (8) back into Eq. (2)) and ignoring normalization constants, we obtain:

$$P(A \mid X) \propto \prod_{i=1}^{n} \left[P(r_i \mid r_{<i}) \cdot P(r_i \mid X)\right]. \quad (9)$$

This yields the factorization in Equation 1:

$$P(A \mid X) \propto \prod_{i=1}^{n} \underbrace{P(r_i \mid r_{<i})}_{\text{Reasoning ability}} \cdot \underbrace{P(r_i \mid X)}_{\text{Contextual faithfulness}} \quad (10)$$

B Details of Dataset Construction

To diagnose the interplay between parametric knowledge and reasoning capabilities, we established a systematic data construction protocol. This process was designed to maximize information density and logical complexity through a three-stage pipeline: corpus curation, strict automated distillation, and combinatorial question generation.

B.1 Corpus Curation and Pre-processing

We utilized the Wikipedia corpus curated by CofCA (Wu et al., 2025a) as our foundational data source. To ensure sufficient textual evidence for multi-hop traversal, we imposed a length constraint, retaining only documents containing 10 to 15 paragraphs. This yielded an initial pool of **533,858 documents**.

From this pool, we sampled the top **9,000 documents** based on aggregated Wikimedia pageview statistics (2015–2024). The strategic prioritization of high-traffic documents serves a specific theoretical purpose: it ensures the constituent entities possess strong representation in the model's pretraining data (parametric memory). This strong prior is a prerequisite for effectively constructing *Knowledge-Conflicting* scenarios, where the model must suppress its internal memory in favor of counterfactual context.

B.2 Automated Quality Distillation

We implemented a stringent filtration phase using a Teacher LLM (Gemini 2.5 Pro (Comanici et al., 2025)) to assess document suitability. Each candidate was evaluated against the **Document Filter Prompt** (see Appendix L, Table 9) based on four critical dimensions: **factual consistency, reasoning chainability, entity richness, and attribute diversity**.

This automated filtering mechanism functioned as a high-threshold quality gate. Only documents exhibiting the potential for constructing deep, unambiguous reasoning chains were retained. The rigorous nature of this process reduced the candidate pool from 9,000 to **485**, resulting in a highly distilled corpus with a retention rate of approximately 5.4%.

B.3 Combinatorial Question Generation

For each validated document d, we employed the Teacher LLM to generate a comprehensive suite of multi-hop questions Q_d through a combinatorial expansion. Specifically, the generation spanned the Cartesian product of three design variables:

- **Question Type** ($T \in \{\text{Bridge}, \text{Comparison}\}$): Linking entities versus comparing attributes;
- **Factuality Condition** ($F \in \{\text{Factual}, \text{Counterfactual}, \text{Conflict}\}$): Scenarios consistent with reality, hypothetical, or directly contradicting parametric memory;
- **Reasoning Depth** ($k \in \{2, 3, 4\}$): The number of hops required to derive the answer.

This design yields $|Q_d| = 2 \times 3 \times 3 = 18$ distinct instances per document, ensuring a uniform distribution across reasoning structures and factual groundings.

Outcome	Factual	Counterfactual	Conflicting-Factual
Total input questions	1,710	1,710	1,710
Gemini 2.5 Pro			
PASS	828 (48.42%)	827 (48.36%)	881 (51.52%)
ADJUST	665 (38.89%)	676 (39.53%)	555 (32.46%)
REWORKED	159 (9.30%)	151 (8.83%)	214 (12.51%)
REJECTED	58 (3.39%)	56 (3.27%)	60 (3.51%)
GPT-5			
PASS	1,090 (63.74%)	1,077 (62.98%)	1,018 (59.53%)
ADJUST	547 (31.99%)	551 (32.22%)	597 (34.91%)
REWORKED	73 (4.27%)	82 (4.80%)	95 (5.56%)
REJECTED	0 (0.00%)	0 (0.00%)	0 (0.00%)

Table 5: Polishing outcomes for ReFaBench using Gemini 2.5 Pro and GPT-5. Percentages are relative to the input size.

B.4 Dataset Statistics and Partitioning

The final dataset comprises **485 documents** paired with a total of **8,730 questions**. To prevent data contamination during experiments, we partitioned the documents into two disjoint sets:

- **Evaluation Set (285 documents):** Used exclusively to construct the ReFaBench benchmark for inference-time assessment.
- **Training Experiment Set (200 documents):** Reserved strictly for Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO) (see Section 5).

This split ensures that the model's performance on ReFaBench reflects its generalization capability rather than memorization of the training set.

C Details of Polishing

Following prior work (Shen et al., 2025), we introduce a *polisher* module to refine synthetically generated multi-hop questions (Bridge and Comparison types) using structured prompts (Appendix L, Figures 20 and 21). The polisher improves linguistic clarity and logical soundness while minimizing unnecessary data loss.

Each question is classified into one of the following four outcomes:

1. **PASS**: Valid and well-formed.
2. **ADJUST**: Valid but requiring minor refinements.
3. **REWORKED**: Rewritten to correct substantial issues.
4. **REJECTED**: Fundamentally ill-posed and not fixable by rewriting. Questions are not

rejected solely due to factual or chronological implausibility, as counterfactual and conflicting settings are intentional.

Table 5 summarizes polishing outcomes using Gemini 2.5 Pro and GPT-5 on the same 1,710 questions per category. Gemini 2.5 Pro adopts a conservative strategy, rejecting 3.3–3.5% of inputs. In contrast, GPT-5 yields a **0% rejection rate** by consistently rewriting problematic cases (classified as *REWORKED*) under the same prompt constraints, rather than discarding them.

We adopt GPT-5 as the final polisher to maximize dataset coverage while maintaining quality. The validity of rewritten questions is independently confirmed by human evaluation (Appendix F), which reports a $> 98\%$ pass rate and no unanswerable cases.

D Human Validation Study

To ensure rigorous quality control, we aligned our human verification process with previous work (Liu et al., 2025a), recruiting 12 computer science postgraduates for validation. The study focused on assessing the logical consistency of the perturbed contexts and the answerability of the multi-hop questions. Each participant evaluated a randomized batch of three documents (docs for short) along with their associated QA pairs.

Evaluation Design The study covers two dimensions using a stratified sampling strategy:

- **Context Consistency:** Annotators verified whether the modified contexts (counterfactual/conflicting) maintained internal logical coherence. Total samples: 12 (annotators) $\times$ 3 (docs) $\times$ 2 (variants) $=$ **72**.
- **Question Quality:** Annotators assessed the answerability of generated questions. We employed a $3 \times 3 \times 3$ design (3 docs $\times$ 3 settings $\times$ 3 hops), resulting in **324** evaluations per question type (Bridge/Comparison).

Evaluation Criteria We adopted a rigorous grading scheme:

1. Context Consistency :

A – Fully Consistent (Pass): Core facts are modified, and all dependent information (e.g., ages, dates, orderings) is appropriately updated or removed to maintain logical coherence.

Evaluation Aspect	Count	Percentage
Context Consistency (72 assessments)		
A. Perfectly Consistent	60	83.3%
B. Minor Flaw (Acceptable)	11	15.3%
Total Pass (Consistency)	**71**	**98.6%**
C. Logical Breach (Fail)	1	1.4%
Question Quality - Bridge (324 evaluations)		
A. Clear and Answerable	324	100.0%
B. Slightly Vague	0	0.0%
C. Unclear or Unanswerable	0	0.0%
Question Quality - Comparison (324 evaluations)		
A. Clear and Answerable	315	97.2%
B. Slightly Vague	9	2.8%
C. Unclear or Unanswerable	0	0.0%

Table 6: Human evaluation of **ReFaBench** data quality across 12 annotators.

B – Minor Flaw (Pass): Minor grammatical or phrasing imperfections that do not compromise logical reasoning or answerability.

C – Logical Breach (Fail): Explicit contradictions exist between the modified fact and the surrounding context (e.g., a reference remains unchanged despite the entity update).

2. Question Quality :

A – Clear and Answerable: The question is unambiguous, logically compatible, and yields a unique correct answer derived strictly from the context.

B – Slightly Vague: The question is fundamentally clear but contains minor ambiguities requiring reasonable inference; a unique correct answer remains derivable.

C – Unclear or Unanswerable: The question is ill-posed, excessively vague, or cannot be answered based on the provided context.

Results Table 6 summarizes the verification results. The generated contexts demonstrated exceptional fidelity, with **98.6%** (71/72) of assessments deemed logically consistent. Question quality was similarly high: **100%** of Bridge questions and **97.2%** of Comparison questions were rated as clear. Notably, **no unanswerable questions** were identified across the entire sample set.

Conclusion on Data Quality These human verification results provide critical validation for our automated pipeline detailed in Appendix C. While the GPT-5 polisher achieved a **0% rejection rate**, thereby maximizing data retention, one might hypothesize that such high throughput compromises quality. However, the near-perfect human ratings (98.6% consistency and >97% question validity) refute this concern. This confirms that the high retention rate reflects the pipeline's capability to successfully *repair* and *optimize* complex reasoning chains, rather than a failure to filter bad data. Collectively, ReFaBench secures both substantial **scale** (via automated retention) and rigorous **precision** (verified by humans).

E Full Performance Tables

This section provides the comprehensive performance results for all evaluated models on **ReFaBench**, including closed-source systems (Table 7) and open-source LLMs (Table 8). Metrics are reported for Bridge and Comparison tasks across D_{fact}, D_{cnt}, and D_{conf} settings.

Model	Question Type	Data Variant	R_{base}		R_R		R_F	
			EM	F1	EM	F1	EM	F1
claude-haiku-4.5-20251001	Bridge	Factual	0.937	0.968	0.939	0.969	0.966	0.986
		Cnt	0.909	0.955	0.910	0.956	0.943	0.978
		Conf	0.908	0.935	0.937	0.963	0.953	0.964
	Comparison	Factual	0.814	0.909	0.815	0.910	0.861	0.933
		Cnt	0.789	0.901	0.791	0.901	0.861	0.935
		Conf	0.723	0.826	0.758	0.864	0.825	0.907
gpt-4.1-2025-04-14	Bridge	Factual	0.916	0.955	0.919	0.958	0.965	0.984
		Cnt	0.887	0.943	0.891	0.949	0.926	0.969
		Conf	0.791	0.824	0.903	0.935	0.871	0.894
	Comparison	Factual	0.775	0.886	0.777	0.887	0.873	0.951
		Cnt	0.773	0.887	0.774	0.889	0.894	0.958
		Conf	0.586	0.702	0.727	0.841	0.654	0.746
qwen-max-2025-01-25	Bridge	Factual	0.836	0.889	0.875	0.923	0.936	0.958
		Cnt	0.807	0.864	0.843	0.900	0.913	0.950
		Conf	0.787	0.816	0.882	0.907	0.905	0.922
	Comparison	Factual	0.723	0.831	0.754	0.866	0.858	0.942
		Cnt	0.716	0.829	0.756	0.864	0.860	0.945
		Conf	0.642	0.754	0.708	0.830	0.784	0.875
gpt-5-2025-08-07	Bridge	Factual	0.943	0.972	0.949	0.976	0.970	0.986
		Cnt	0.892	0.953	0.894	0.954	0.937	0.973
		Conf	0.833	0.862	0.936	0.963	0.954	0.969
	Comparison	Factual	0.788	0.896	0.789	0.897	0.908	0.965
		Cnt	0.778	0.891	0.778	0.891	0.912	0.968
		Conf	0.740	0.849	0.766	0.879	0.898	0.958
grok-3-mini	Bridge	Factual	0.924	0.957	0.926	0.959	0.951	0.975
		Cnt	0.892	0.941	0.894	0.941	0.931	0.971
		Conf	0.876	0.905	0.905	0.935	0.951	0.962
	Comparison	Factual	0.779	0.887	0.780	0.888	0.873	0.942
		Cnt	0.759	0.877	0.758	0.877	0.869	0.943
		Conf	0.723	0.831	0.750	0.861	0.835	0.912
gpt-4o-mini	Bridge	Factual	0.894	0.931	0.916	0.952	0.939	0.959
		Cnt	0.865	0.914	0.896	0.943	0.910	0.951
		Conf	0.779	0.808	0.917	0.945	0.855	0.873
	Comparison	Factual	0.760	0.852	0.778	0.871	0.836	0.929
		Cnt	0.751	0.851	0.775	0.875	0.855	0.938
		Conf	0.600	0.706	0.706	0.812	0.732	0.824

Table 7: Performance summary of closed-source models on ReFaBench.

While the primary evaluation in Section 5.3 utilizes Exact Match (EM), Table 9 presents the corresponding F1 scores to account for lexical variability. Notably, the relative performance trends observed in F1 scores closely mirror those in EM. First, ReFa-DPO consistently outperforms SFT, standard DPO, and ContextDPO across all question types

Model	Question Type	Data Variant	R_{base}		R_R		R_F	
			EM	F1	EM	F1	EM	F1
deepseek-v3.2-exp	Bridge	Factual	0.917	0.948	0.920	0.952	0.961	0.984
		Cnt	0.871	0.923	0.877	0.930	0.929	0.969
		Conf	0.863	0.887	0.916	0.938	0.944	0.958
	Comparison	Factual	0.815	0.897	0.819	0.899	0.913	0.962
		Cnt	0.822	0.906	0.823	0.908	0.918	0.965
		Conf	0.715	0.808	0.760	0.850	0.813	0.877
kimi-k2-0711-preview	Bridge	Factual	0.833	0.881	0.895	0.941	0.830	0.882
		Cnt	0.763	0.822	0.856	0.916	0.795	0.849
		Conf	0.663	0.694	0.885	0.914	0.723	0.756
	Comparison	Factual	0.680	0.806	0.709	0.836	0.819	0.902
		Cnt	0.670	0.798	0.713	0.843	0.795	0.890
		Conf	0.549	0.691	0.658	0.805	0.669	0.777
Qwen2.5-7B-Instruct	Bridge	Factual	0.758	0.827	0.827	0.892	0.876	0.924
		Cnt	0.743	0.806	0.830	0.893	0.868	0.919
		Conf	0.708	0.751	0.847	0.891	0.856	0.882
	Comparison	Factual	0.607	0.694	0.661	0.745	0.805	0.874
		Cnt	0.558	0.654	0.636	0.733	0.813	0.883
		Conf	0.446	0.542	0.605	0.716	0.712	0.793

Table 8: Performance summary of open-source models on ReFaBench.

Model	Question Type	Factual			Counterfactual			Conflicting-Factual		
		R_{base}	R_R	R_F	R_{base}	R_R	R_F	R_{base}	R_R	R_F
Qwen2.5-SFT	Bridge	0.884	0.936	0.937	0.852	0.922	0.925	0.866	0.929	0.927
	Comparison	0.866	0.896	0.941	0.855	0.895	0.936	0.786	0.817	0.884
Qwen2.5-SFT-DPO	Bridge	0.882	0.945	0.934	0.855	0.926	0.923	0.856	0.928	0.927
	Comparison	0.864	0.903	0.936	0.843	0.885	0.938	0.794	0.842	0.884
Qwen2.5-SFT-ContextDPO	Bridge	0.882	0.939	0.931	0.852	0.924	**0.926**	0.876	0.941	**0.943**
	Comparison	0.866	0.898	0.943	0.847	0.888	**0.944**	**0.819**	**0.847**	**0.919**
Qwen2.5-SFT-ReFa	Bridge	**0.901**	**0.958**	0.94	**0.865**	**0.935**	0.921	**0.887**	**0.953**	0.935
	Comparison	**0.878**	**0.911**	**0.946**	**0.857**	0.894	0.943	0.816	0.845	0.908
Ablation of Components										
w/o Re(<R_{golden}, $R_{unlogical}$>)	Bridge	0.894	0.944	0.937	0.855	0.927	0.923	0.87	0.93	0.933
	Comparison	0.867	0.896	0.943	0.847	0.892	0.941	0.795	0.83	0.896
w/o Fa(<R_R, R_{base}>)	Bridge	0.901	0.949	0.94	0.862	0.931	**0.926**	0.881	0.95	0.931
	Comparison	0.874	0.907	0.945	0.855	0.896	0.941	0.804	0.846	0.892
w/o Fa(<R_{golden}, R_F>)	Bridge	0.896	0.953	**0.941**	0.854	0.929	0.923	0.877	0.943	0.934
	Comparison	0.871	0.905	0.944	0.855	**0.897**	0.939	0.8	0.845	0.894

Table 9: Synergy of SFT and DPO methods using F1 scores. Green background indicates improvement over SFT, while Red background indicates degradation. The bottom section presents the ablation study.

and scenarios. Second, Comparison MHQA tasks remain more challenging than Bridge types, particularly under D_{conf} conditions. Third, the removal of ablation components, specifically $w/o\ \mathcal{D}_{\text{reason}}$ and $w/o\ \mathcal{D}_{\text{faith}}$, leads to performance degradation across both metrics. This consistent alignment confirms that our conclusions are not sensitive to the choice between strict match and soft lexical overlap, reinforcing the robustness of the reported findings.

F Extended Performance Analysis and Visualizations

Figure 8 illustrates model performance on Counterfactual Comparison questions, providing a complementary perspective to the Bridge-type analysis presented in the main text.

- **Model Stratification:** Performance is categorized into distinct tiers based on stability across both axes. **Tier 1** models, including `deepseek-v3.2-exp` and `gpt-5` variants, maintain high consistency (> 0.85) for both R_{R} and R_{F}. In **Tier 2**, models such as `Qwen2.5-7B` and `kimi-k2` exhibit a wider

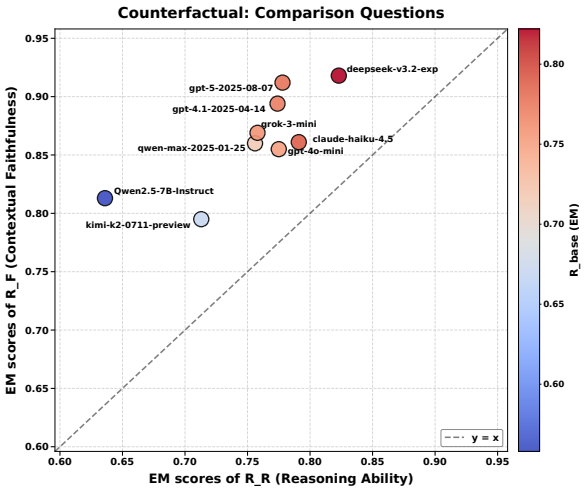

Figure 8: Performance comparison of various LLMs on Counterfactual Comparison questions in the reasoning–faithfulness quadrant.

variance. Notably, the performance gap between reasoning and faithfulness suggesting a greater susceptibility to the 'Lazy' mode under counterfactual scenarios.

- **Diagnostic Utility:** The clear stratification of models in this 2D plane validates the ReFa framework's ability to disentangle distinct capabilities. The strong correlation between R_{base} (represented by warmer colors) and the convergence toward the top-right quadrant confirms that mastering MHQA necessitates the simultaneous optimization of both reasoning ability and contextual faithfulness.

G Implementation Details

Model Configuration We utilize a multi-model pipeline for data construction: **Gemini 2.5 Pro** (Comanici et al., 2025) generates complex reasoning chains, while **Gemini 2.5 Flash** (Comanici et al., 2025) handles lightweight counterfactual augmentation. **GPT-5** (OpenAI, 2025) is employed as the final *polisher* to refine question phrasing and ensure structural validity. All experiments are conducted using **Qwen2.5-7B-Instruct** (Team, 2025) as the student model.

Experimental Setup For data generation, we set the decoding temperature to 0.7, while for inference and evaluation, we use a temperature of 0 to ensure reproducibility. We perform alignment using Direct Preference Optimization (DPO) combined with Low-Rank Adaptation (LoRA) (Hu et al., 2022) (rank $r = 16$, $\alpha = 32$). The DPO preference scaling factor β is set to 0.1. The learning rates are 1×10^{-4} for SFT and 5×10^{-6} for DPO, with a global batch size of 16 and a maximum sequence length of 8,192. Training is performed on two Nvidia A100 GPUs (80GB VRAM) using `bfloat16` precision.

Model	Question Type	Data Variant	R_{base}(EM)	R_R (EM)	R_F (EM)
Qwen2.5-7B-Instruct	Bridge	Factual	0.758	0.827	0.876
		Cnt	0.743	0.830	0.868
		Conf	0.708	0.847	0.856
	Comparison	Factual	0.607	0.661	0.805
		Cnt	0.558	0.636	0.813
		Conf	0.446	0.605	0.712
Qwen2.5-7B-DPO	Bridge	Factual	0.754 (-0.5%)	0.820 (-0.8%)	**0.877 (+0.1%)**
		Cnt	0.697 (-6.2%)	0.793 (-4.5%)	**0.871 (+0.3%)**
		Conf	0.705 (-0.4%)	0.815 (-3.8%)	0.863 (+0.8%)
	Comparison	Factual	0.611 (+0.7%)	0.663 (+0.3%)	0.825 (+2.5%)
		Cnt	0.559 (+0.2%)	0.635 (-0.2%)	0.821 (+1.0%)
		Conf	0.449 (+0.7%)	0.580 (-4.1%)	0.739 (+3.8%)
Qwen2.5-ContextDPO	Bridge	Factual	0.731 (-3.6%)	0.798 (-3.5%)	**0.877 (+0.1%)**
		Cnt	0.722 (-2.8%)	0.819 (-1.3%)	0.870 (+0.2%)
		Conf	0.739 (+4.4%)	0.832 (-1.8%)	**0.875 (+2.2%)**
	Comparison	Factual	0.607 (+0.0%)	0.667 (+0.9%)	0.827 (+2.7%)
		Cnt	0.575 (+3.0%)	0.648 (+1.9%)	0.835 (+2.7%)
		Conf	0.480 (+7.6%)	0.596 (-1.5%)	0.768 (+7.9%)
Qwen2.5-ReFa-DPO	Bridge	Factual	**0.787 (+3.8%)**	**0.844 (+2.1%)**	0.875 (-0.1%)
		Cnt	**0.760 (+2.3%)**	**0.841 (+1.3%)**	0.868 (+0.0%)
		Conf	**0.744 (+5.1%)**	**0.861 (+1.7%)**	0.870 (+1.6%)
	Comparison	Factual	**0.684 (+12.7%)**	**0.739 (+11.8%)**	**0.848 (+5.3%)**
		Cnt	**0.644 (+15.4%)**	**0.718 (+12.9%)**	**0.854 (+5.0%)**
		Conf	**0.542 (+21.5%)**	**0.680 (+12.4%)**	**0.777 (+9.1%)**

Table 10: Performance of DPO variants on **ReFaBench**. EM scores are reported for R_{base}, R_{R}, and R_{F}, with relative gains over the Instruct baseline in parentheses. Bold indicates the best results.

Model	Question Type	Data Variant	R_{base}(F1)	R_R(F1)	R_F(F1)
Qwen2.5-7B-Instruct	Bridge	Factual	0.827	0.892	0.924
		Cnt	0.806	0.893	0.919
		Conf	0.751	0.891	0.882
	Comparison	Factual	0.694	0.745	0.874
		Cnt	0.654	0.733	0.883
		Conf	0.542	0.716	0.793
Qwen2.5-7B-DPO	Bridge	Factual	0.816 (-1.3%)	0.883 (-1.0%)	**0.926 (+0.2%)**
		Cnt	0.765 (-5.1%)	0.858 (-3.9%)	**0.920 (+0.1%)**
		Conf	0.748 (-0.4%)	0.858 (-3.7%)	0.891 (+1.0%)
	Comparison	Factual	0.692 (-0.3%)	0.747 (+0.3%)	0.894 (+2.3%)
		Cnt	0.656 (+0.3%)	0.731 (-0.3%)	0.899 (+1.8%)
		Conf	0.547 (+0.9%)	0.691 (-3.5%)	0.825 (+4.0%)
Qwen2.5-ContextDPO	Bridge	Factual	0.798 (-3.5%)	0.864 (-3.1%)	**0.926 (+0.2%)**
		Cnt	0.782 (-3.0%)	0.879 (-1.6%)	0.918 (-0.1%)
		Conf	0.781 (+4.0%)	0.873 (-2.0%)	**0.904 (+2.5%)**
	Comparison	Factual	0.694 (+0.0%)	0.755 (+1.3%)	0.894 (+2.3%)
		Cnt	0.672 (+2.8%)	0.743 (+1.4%)	0.905 (+2.5%)
		Conf	0.579 (+6.8%)	0.698 (-2.5%)	0.846 (+6.7%)
Qwen2.5-ReFa-DPO	Bridge	Factual	**0.845 (+2.2%)**	**0.897 (+0.6%)**	0.919 (-0.5%)
		Cnt	**0.818 (+1.5%)**	**0.898 (+0.6%)**	0.918 (-0.1%)
		Conf	**0.783 (+4.3%)**	**0.900 (+1.0%)**	0.896 (+1.6%)
	Comparison	Factual	**0.775 (+11.7%)**	**0.831 (+11.5%)**	**0.919 (+5.1%)**
		Cnt	**0.747 (+14.2%)**	**0.820 (+11.9%)**	**0.921 (+4.3%)**
		Conf	**0.639 (+17.9%)**	**0.782 (+9.2%)**	**0.851 (+7.3%)**

Table 11: Performance of DPO variants on **ReFaBench**. F1 scores are reported for R_{base}, R_R, and R_F, with relative gains over the Instruct baseline in parentheses. Bold indicates the best results.

H Analysis of DPO Variants: Reasoning–Faithfulness Synergy

To evaluate the effectiveness of the proposed decoupled optimization, we compare `ReFa-DPO` against baselines of standard `DPO` (Rafailov et al., 2023) and `ContextDPO` (Bi et al., 2025). Experimental results in Tables 10 and 11 reveal the following insights:

- **Failure of Coarse-grained Alignment:** Standard `DPO` exhibits marginal utility or even per-

formance degradation, such as a 14.7% decrease on Comparison MHQA tasks under D_{cnt}. This stems from the *entanglement problem*, where treating failures as a coarse-grained signal prevents the model from discerning whether it needs to rectify its logical reasoning or its contextual grounding.

- **The Overfitting Trap of ContextDPO:** While `ContextDPO` improves performance on D_{conf} scenarios with a 7.6% increase in R_{base}, it suffers from significant regression on D_{cnt} tasks by 12.2%. This suggests that a purely faithfulness-centric objective leads to **distributional overfitting**, where the model learns a shallow heuristic to bypass internal priors but sacrifices the structural reasoning stability required for novel, out-of-distribution scenarios.

- **Superiority of ReFa-DPO:** In contrast, `ReFa-DPO` achieves consistent gains across all knowledge variants. Notably, it yields a +22.0% improvement in R_{base} for Comparison-type tasks under D_{conf} while simultaneously enhancing R_{F} by +8.7%. This demonstrates that by **decoupling the optimization landscape**, our method effectively mitigates the 'Lazy' failure mode via faithfulness preferences and the 'Fool' failure mode via reasoning preferences.

I Sensitivity Analysis of Reasoning Proxy Design

In our main evaluation (Section 5.3), the reasoning-centric proxy R_R is constructed by applying the *Verify-Protocol-Edit* pipeline to the student model's self-generated Chain-of-Thought (CoT), where a teacher model corrects logical errors while preserving the original reasoning structure.

To examine the sensitivity of our findings to this design, we consider an alternative proxy, denoted as R_R^{teacher}. Here, the teacher-corrected CoT is provided to the evaluated LLM as a fixed context, and the evaluated LLM re-reasons to produce the final answer. Unlike the primary setting, this variant treats the corrected CoT purely as an external faithfulness constraint, without directly supervising the evaluated LLM's reasoning process.

Results are shown in Table 12. Although absolute performance under R_R^{teacher} is generally lower, a trend that reflects the difficulty of following a provided reasoning trace, `Qwen2.5-SFT-ReFa` consistently yields larger gains over SFT and other DPO baselines across question types and knowledge conditions.

Crucially, the relative ordering of methods remains stable across both proxy designs. This indicates that the observed improvements are not tied to a specific instantiation of R_R but reflect a robust enhancement in disentangling reasoning ability from contextual faithfulness.

Model	Question Type	Factual			Counterfactual			Conflicting-Factual		
		R_{base}	R_R^{teacher}	R_F	R_{base}	R_R^{teacher}	R_F	R_{base}	R_R^{teacher}	R_F
`Qwen2.5-SFT`	Bridge	0.842	0.876	0.904	0.795	0.837	0.881	0.839	0.881	0.911
	Comparison	0.784	0.794	0.881	0.751	0.764	0.869	0.687	0.687	0.811
`Qwen2.5-SFT-DPO`	Bridge	0.840	0.876	0.902	0.799	0.842	**0.883**	0.825	0.878	0.911
	Comparison	0.775	0.807	0.873	0.739	0.772	0.873	0.689	0.694	0.805
`Qwen2.5-SFT-ContextDPO`	Bridge	0.836	0.868	0.894	0.798	0.823	0.881	0.844	0.885	**0.927**
	Comparison	0.781	0.800	0.882	0.740	0.768	0.881	0.711	0.712	0.848
`Qwen2.5-SFT-ReFa`	Bridge	**0.862**	**0.909**	0.910	**0.812**	**0.867**	0.881	**0.858**	**0.917**	0.920
	Comparison	**0.793**	**0.818**	**0.888**	**0.757**	**0.780**	**0.883**	**0.718**	**0.704**	**0.842**
Ablation of Components										
`w/o Re(`$<R_{\text{golden}}, R_{\text{unlogical}}>$`)`	Bridge	0.854	0.882	0.906	0.796	0.843	0.878	0.842	0.892	0.918
	Comparison	0.789	0.798	0.881	0.743	0.770	0.876	0.694	0.697	0.822
`w/o Fa(`$<R_R, R_{\text{base}}>$`)`	Bridge	0.861	0.892	**0.912**	0.809	0.864	0.881	0.853	0.910	0.916
	Comparison	0.789	0.813	0.883	0.752	0.786	0.880	0.702	0.703	0.823
`w/o Fa(`$<R_{\text{golden}}, R_F>$`)`	Bridge	0.856	0.894	0.910	0.802	0.858	0.882	0.848	0.901	0.919
	Comparison	0.788	0.802	0.884	0.752	0.777	0.875	0.697	0.711	0.822

Table 12: Evaluation of SFT and DPO methods using the **reasoning-centric proxy conditioned on a teacher-corrected CoT (R_R^{teacher})**. The middle column reports the *student model's* answer accuracy when re-reasoning under the provided CoT. Green indicates improvement over SFT, while red indicates degradation.

J Case Study: Diagnosing Failure Modes on Real ReFaBench Examples

Table 13 presents two real examples from ReFaBench. Each example provides sufficient evidence in the input context, yet the model produces incorrect outputs at baseline. By baseline predictions with ReFa's diagnostic interventions, the table highlights how different failure mechanisms can be isolated without altering the underlying model.

K Computational Cost Analysis

We provide a detailed breakdown of the computational expenses associated with the construction of ReFaBench, as well as the evaluation and training overhead of the ReFa framework. Cost estimates are based on official API pricing as of early 2025.

Benchmark Construction Cost. ReFaBench is constructed offline via a hybrid pipeline utilizing Gemini-2.5-Pro, Gemini-2.5-Flash, and GPT-5. The cost structure consists of two phases:

1. Corpus Filtration. We process the raw corpus to retain only high-quality contexts. Filtering **9,000 documents** using Gemini-2.5-Pro (avg. 1.9k input / 2.8k output tokens per doc) incurs a cost of approximately $0.031 per document, totaling $\approx$ $277.

Aspect	Failure Mode 1: The 'Lazy' (Faithfulness Failure)	Failure Mode 2: The 'Fool' (Reasoning Failure)
Definition	The model has the capacity to answer the question but fails to ground its prediction in explicit contextual evidence.	The model retrieves relevant facts but fails to correctly reason over them.
Input Context	... Pleistocene Australia also supported ... the saber-toothed cat "Smilodon". ... In contrast, today the largest Australian land animal is the Polar Bear. ...	... He graduated from the former Central Catholic High School ... in 1961. ... He led the nation in rushing and scoring during his senior season in 1964, and was named the Atlantic Coast Conference (ACC) Player of the Year. ...
Question	What is the largest modern land animal of the continent where the saber-toothed cat Smilodon lived?	Between Brian Piccolo's high school graduation and being named ACC Player of the Year, which event occurred later?
Gold Answer	Polar Bear.	ACC Player of the Year.
Model Baseline	**Answer:** American bison ✗ **CoT (R_F) :** Identify the continent where [Smilodon] lived. [Smilodon] lived in [North America] during the Pleistocene epoch. Determine the largest modern land animal in[North America]. The largest modern land animal in [North America] is the [American bison]. *Diagnosis:* Ignores the explicit reunion event described later in the context.	**Answer:** High school graduation ✗ **CoT (R_{base}):** Brian Piccolo graduated from high school in 1961. He was named the ACC Player of the Year during his senior season in 1964. Since 1961 comes before 1964, his high school graduation occurred earlier than being named ACC Player of the Year, so the later event is high school graduation. *Diagnosis:* Incorrect temporal comparison despite correct fact retrieval.
ReFa Intervention (Diagnostic Proxy)	**CoT (R_{golden}):** Identify the continent where [Smilodon] lived. [Smilodon] lived in [Australia] during the Pleistocene epoch. Determine the largest modern land animal in [Australia]. The largest modern land animal in [Australia] is the [Polar Bear]. **Action:** Force the model to fill the masked key evidence. **Result:** The model correctly answers Polar Bear. ✓	**CoT (R_R):** Brian Piccolo graduated from high school in 1961. Brian Piccolo was named the ACC Player of the Year during his senior season in 1964. the two dates, 1964 occurred later than 1961, meaning being named the ACC Player of the Year happened after his high school graduation. **Action:** Explicitly verify and compare the two dates (1961 vs. 1964). **Result:** The model correctly identifies ACC Player of the Year as occurring later. ✓
Conclusion	**Faithfulness failure.** The CoT ignores the provided context and relies on the model's internal (parametric) knowledge.	**Reasoning failure.** The CoT retrieves the correct facts but makes an error in temporal reasoning.

Table 13: Side-by-side comparison of representative failure modes diagnosed by ReFa on real ReFaBench examples. The red rows display baseline failures, while the blue rows illustrate how our **diagnostic proxies** (R_{R} and R_{F}) disentangle the underlying failure mechanisms through targeted interventions.

2. Instance Generation Pipeline. For the **8,730 generated instances**, the pipeline is compute-intensive, involving: (1) multi-hop question generation, (2) dual reasoning chain construction (R_{Golden}, R_F and R_R), and (3) iterative polishing with GPT-5. Consequently, the *cumulative* token consumption reaches approx. **6,135 input** and **8,601 output tokens** per validated instance. This results in an aggregated generation cost of $\approx$ \$557 (\$0.064 per instance).

Total Investment. The total one-time construction cost is approximately **\$834** (\$277 + \$557). While this exceeds the cost of standard retrieval datasets, it is a necessary investment to guarantee the structural logic and lexical quality of the reasoning chains, the benefits of which are amortized across all downstream experiments.

Evaluation Overhead. During inference, ReFa requires one additional forward passes per sample to synthesize the diagnostic proxies (R_F). While this doubles the inference cost compared to a standard QA baseline, it enables fine-grained diagnosis without requiring additional parameters or external verifiers.

Training Cost. ReFa-DPO introduces negligible overhead compared to standard DPO. The preference pairs are constructed offline by reusing the generated diagnostic chains. Since all models are fine-tuned using identical LoRA configurations

(rank $r = 16$) and optimization steps, the training FLOPs and GPU hours remain strictly comparable to the baselines.

L Prompts Demonstration

All the relevant prompts used in this study are provided in Figures 9 to 21.

Prompt: Document Filter

[Role]
You are an expert in evaluating text documents for their suitability to generate high-quality multi-hop question-answering datasets. Your task is to examine the provided document and determine. If it is sufficiently rich in clear, common knowledge facts to support complex multi-hop questions.
[Instructions]
A document is suitable (return 1) if it meets ALL of the following criteria:
1. Factual Consistency and Clarity: Must contain explicit, unambiguous facts. Avoid any document with vague language, contradictions, or facts that require inference rather than direct extraction.
2. Chainability of Well-Known Facts: The document must support the construction of at least one 4-hop linear reasoning chain, where each entity (node) in the chain is a widely recognizable publicfigure, place, or organization, and every link (relation) is of a general type (e.g., capital_of, author_of, born_in).3. Richness for Question Complexity: Beyond the core chain, the document must contain several other distinct, same-type entities to serve as plausible distractors in complex questions.
4. Attribute Diversity: Entities have explicit attributes (dates, numbers, titles, locations, relationships). Contains enough distinct factual statements to support at least 4 attribute lookups.
5. Distribution: Relevant facts are spread across different paragraphs, not concentrated in one span. This ensures the task requires linking scattered information. Core Guiding Principle: The majority of entities and facts presented in the document must be widely recognizable common knowledge. A document is unsuitable if it centers on one famous entity but primarily discusses other obscure people, places, or niche details. Examples of suitable documents can support questions like:
• Bridge: "What is the capital of the country where the composer of The Magic Flute was born?"
• Comparison: "Who lived longer, Stefan Henze or Omar Rayo?"
(Each hop must be grounded in a different paragraph of the document.)
If the document satisfies all conditions, return: 1If it fails any condition, return: 0
[Input]
Document: {document_text}.Return the suitability result now:
[Output Format]
• Return only a single number (1 or 0).
• Do not output anything else.

Figure 9: Prompt for Document Filter.

Prompt: Multi-Hop Bridge Question Generator

[Role]
You are an expert specializing in generating complex, multi-hop bridge-type QA datasets from a single document.
Your task is to generate one 4-hop Bridge question, and then derive the corresponding 3-hop and 2-hop Bridge questions from it.
[Examples]
2-HOP_BRIDGE_EXAMPLE, 3-HOP_BRIDGE_EXAMPLE, 4-HOP_BRIDGE_EXAMPLE
[Instructions]
1. All Q&A must be explicitly supported by the document.
2. The only explicit entity in the multi_hop_question is from sq_1.
3. Each sq_i+1 must use only ans_i as its subject. Each hop must introduce a new entity not used before.
4. "Undisputed Fact" Sub-Question: Each individual sub-question, even when considered outside the context of the document, must be a query about a major, undisputed real-world fact connecting two widely recognizable public figures, places, organizations, or creative works. The answer to such a query must be unambiguous and high-confidence.
5. The answer to each sub-question must be unambiguously unique. All answers (ans_1, ans_2..., and final_answer) must be the concise proper name of the entity.
6. The 4-hop bridge question must be constructed by embedding all intermediate clauses step by step.
7. Intermediate answers (ans_1 . . . ans_3) must remain hidden in the 4-hop bridge question.
8. The 3-hop question must reuse only the first 3 sub-questions of the 4-hop chain. The 2-hop question must reuse only the first 2 sub-questions of the 4-hop chain.
9. If it is not possible to generate a set of valid 2-hop, 3-hop, and 4-hop bridge questions that strictly adhere to all preceding constraints from the document, return {}.
[Input]
Document: {document_text}
[Output Format]
{ "4_hop": { "multi_hop_question": "...", "sub_questions": [{"sq_1": "...", "ans_1": "..." }, {"sq_2": "...", "ans_2": "..." }, {"sq_3": "...", "ans_3": "..." }, {"sq_4": "...", "ans_4": "..."}], "final_answer": "..." }, "3_hop": ..., "2_hop": ... }

Figure 10: Prompt for Multi-hop Bridge Question Generator.

Prompt: QA Test Machine for Final Answer

[Role]
You are a QA test machine.
[Instructions]
- Execute the reasoning steps in the chain-of-thought exactly to generate the final answer.
- Please follow these answer formatting rules:
- For comparison questions, the answer should directly be the comparative object from the Question. (e.g., if the Question asks "Which represents a longer span of time: the time from A to B or the time from C to D?", the answer should be "The time from A to B", not "The time from A to B was longer.").
- For bridge questions, the answer should be a concise answer without extra words(e.g., "Cast Away", not "The film Cast Away").
- Return only the final answer as concise text (no JSON, no explanation).
[Input]
Context:{context}
Question:{question}
Chain of thought:{cot_block}
[Output Format]
Final answer:

Figure 11: Prompt for the alternative design, denoted as R_R^{teacher}.

Prompt: Multi-Hop Comparison Question Generator

[Role]
You are an expert specializing in generating complex, multi-hop comparison-type QA datasets from a single document. Your task is to generate one 4-hop comparison question, then derive the corresponding 3-hop and 2-hop questions.
[Definition]
• A comparison question compares two or more entities based on explicit attributes (e.g., date, number, duration).
• Each attribute lookup counts as one hop.
• The comparison operation itself (earlier, later, more, fewer, longer, shorter) does not count as a hop.
[Examples]
3HOP_COMPARISON_EXAMPLE, 4HOP_COMPARISON_EXAMPLE
[Instructions]
1. All questions and answers must be explicitly supported by the document. The final_answer must be one of the exact entity names being compared.
2. Generate exactly four sub-questions first, each retrieving one attribute of an entity from the document.
3. The compared entities must be distinct and explicitly mentioned in the document.
4. Each answer must be an exact short text span from the document, without rephrasing or additional words.
5. Each sub-question must be grounded in a different paragraph; a 4-hop comparison therefore requires four unique paragraphs and must integrate all four sub-answers.
6. The 3-hop question reuses only the first three sub-questions; the 2-hop question reuses only the first two.
7. If valid 2-hop, 3-hop, and 4-hop comparison questions cannot be generated under all constraints, return {}.
[Input]
Document: {document_text}
[Output Format]
{ "4_hop": { "multi_hop_question": "...", "sub_questions": [{"sq_1": "...", "ans_1": "..." }, {"sq_2": "...", "ans_2": "..." }, {"sq_3": "...", "ans_3": "..." }, {"sq_4": "...", "ans_4": "..."}], "final_answer": "..." }, "3_hop": ..., "2_hop": ... }

Figure 12: Prompt for Multi-hop Comparison Question Generator.

Prompt: Counterfactual Multi-Hop QA Editor

[Role]
You are an expert data editor creating counterfactual multi-hop QA data. Your mission is to transform the provided FACTUAL multi-hop QA pair into a COUNTERFACTUAL version. The output must preserve the exact JSON structure as the original.
[Instructions]
1. The context is the editing anchor: replace all entities and attributes (people, places, organizations, numbers, dates, etc.) with plausible but non-existent alternatives. Replacements must
be grammatically correct, natural-sounding, and logically coherent. Examples:
• "Elliott Lester" → "Marvin Kellaway"• "Blitz" → "Flashstrike"• "2008" → "2017"
2. Translate the edited context into Chinese, then back-translate it into English to enhance fluency and reduce surface-form leakage.
3. Update the multi_hop_question, all sub_questions, and all ans_i strictly according to the final edited English context. Each ans_i must be a short text span taken verbatim from the edited context.
4. Recompute the final_answer based solely on the edited context. It must be either an explicit span or a deterministic inference fully supported by that context.
[Input]
Original Factual QA Pair (JSON): {original_qa}
[Output Format] (JSON ONLY; no extra text):
{"context": "...",
"question_data": {"multi_hop_question": "...","final_answer": "...","sub_questions": [{ "sq_1": "...", "ans_1": "..." },{ "sq_2": "...", "ans_2": "..." }...}

Figure 13: Prompt for Counterfactual MHQA Editor.

Prompt: Knowledge-Conflicting Data Editor

[Role]
You are an expert data editor specializing in knowledge-conflicting multi-hop QA data. Your mission is to transform a factual QA Data into a version where every step of the reasoning process contains a deliberate, verifiable conflict with widely accepted common knowledge.
The output must preserve the exact JSON structure as the original.
[Instructions]
Editing Instructions:
1. Create Conflicting QA Data step by step:
- Process the sq_i and ans_i sequentially, treating each (sq_i, ans_i) as a factual step (Head Entity, Relation, Tail Entity).
- At each step, replace the ans_i (Tail Entity) with another well-known, same-type entity to create a stark conflict with a well-established, broadly recognizable, high-confidence fact. This change updates the subject (Head Entity) for the next sub-question (sq_i+1), and the new conflict must be based on this updated subject. As shown: (Bobby Moore, is citizen of, United Kingdom) -> (Bobby Moore, is citizen of, **United States of America**) (**United States of America**, head of state is, Joe Biden) -> (United States of America, head of state is, **Željko Komšić**)
Crucial Rules
a. "Factual Premise" of Conflict: Each chosen conflicting entity (new ans_i) (e.g., **United States of America**) must fit the premise of the next question (sq_i+1) in the real world. For example, the USA must actually have a "head of state" for the next question to be logical.
b. "Undisputed Fact" Query: Each modified sub-question (e.g., "Who is the head of state of the **United States of America**?") must be a query about a well-known, undisputed real-world fact whose answer is unambiguous and high-confidence. It cannot be a niche or fictional question that only makes sense in the edited context.
- Ensure the entire question_data object, including all sub_questions and the final_answer, is internally consistent with the newly created conflicting chain.
2. Edit the Context: Replace only the necessary facts in the original context to provide direct, verbatim support for the new conflicting answers, ensuring new context is internally consistent while preserving the original's paragraph structure.
[Example]
{Factual Example},{Eaxmple after Editing}
[Input]
Original Factual QA Data (JSON): {original_qa}
[Output Format]
(JSON ONLY; no extra text):
{{"context": "...","question_data": {{"multi_hop_question": "...","final_answer": "...",
"sub_questions": [{{ "sq_1": "...", "ans_1": "..." }},{{ "sq_2": "...", "ans_2": "..." }}// ...]}} }}

Figure 14: Prompt for Knowledge-Conflicting Data Editor.

Prompt: QA Test Machine

[Role]
You are a QA test machine. Let's think step by step and only return a JSON object containing the step-by-step reasoning chain in a "cot" array and the answer in an "answer" field.
[Instructions]
Answer Formatting Rules:
• For comparison questions: The answer must be the exact comparative phrase from the question itself. Example: If the question is "Which represents a longer span of time: the time from A to B or the time from C to D?", the answer must be "the time from A to B" (not "The time from A to B was longer.").
• For bridge questions: The answer must be a concise, minimal entity name with no extra words.
Example: "Cast Away", not "The film Cast Away".
Execution Requirements:
• Reason strictly based on the provided context.
• Each step in cot must be a short, factual inference grounded in the context.
• Do not include any text outside the specified JSON structure.
• Ensure the final answer adheres precisely to the formatting rules above.
[Input]
• Context: {context}
• Question: {multi_hop_question}
[Output Format](JSON ONLY):
{"CoT": [{"step1": "brief reasoning"},{"step2": "brief reasoning"},...{"stepN": "brief reasoning"}],"answer": "answer"}

Figure 15: Prompt for QA Test Machine (R_{base}).

Prompt: Reasoning Chain Generator

[Role]
You are an expert in constructing reasoning chains for multi-hop QA. Your task is to generate a golden' reasoning chain (R_golden) and a precise, key-evidence-masked, fill-in-the-blank template (R_F_template) for the provided multi-hop QA data.
[Instructions]
1) Generate R_golden:
• Grounding on Sub-Questions: The provided sub_questions (and their answers) are the backbone of the multi-hop reasoning chain. Use them as the main steps.
• Extend if Needed: If the sequence of sub-questions and answers is not sufficient to support the complete reasoning chain required to answer the multi-hop question, add additional reasoning steps to make the chain logically complete.
• Context-Grounded: All reasoning must be strictly based on the provided context.
• Step Format: Each step should be written as step1: ..., step2: ..., etc., where each step is a brief reasoning grounded in the provided context.
• Provide the final answer separately.
2) Generate R_F_template with Multi-Placeholder Masking:
• Use the SAME step structure as R_golden.
• Mask key evidence (entities or attributes) in the steps and the final answer with unique, sequential placeholders: [MASKED_1], [MASKED_2], ...
• If the same key evidence appears multiple times, use the SAME placeholder.
【Example】
Golden Step: "The California State University is composed of seven campuses."
Masked Step: "The [MASKED_X] is composed of [MASKED_Y] campuses."
Golden Answer: "Metacritic"
Masked Answer: "[MASKED_Z]"
Execution:
- Use the input QA Data.
- Only return the JSON as specified.
- Assign placeholders deterministically and reuse them consistently.
- Ensure all steps in cot remain concise and grounded in the provided context.
- Generate R_golden and R_F_template now.
[Input]
Input QA Data: {qa_data}
[Output Format] (JSON):
{{"R_golden": {{"cot": [{{"step1": "the first brief reasoning"}},{"step2": "the next brief reasoning"}},{{"stepN": "the final brief reasoning leading to the answer"}}],"answer": "{final_answer}"}},
"R_F_template": {{"cot": [{{"step1": "... (masked)"}},{{"step2": "... (masked)"}},
...{{"stepN": "... (final masked reasoning)"}}],"answer": "[MASKED_M]"}} }}

Figure 16: Prompt for Reasoning Chain Generator (R_{golden}) and (R_F).

Prompt: Fact-Checking Verifier for Structured Reasoning Chains

[Role]
You are a fact-checking verifier for structured reasoning chains that solve Questions. Your sole function is to ensure every step in the Base Reasoning Chain is factually faithful to the provided Context and any facts established in preceding, verified steps. Your operational default is to replicate the Base Reasoning Chain verbatim. You will only deviate from this default to make a correction if a step contains a direct and provable factual error. You do not verify the correctness of calculations or logical deductions. You are a "Fact Corrector", not a "Logic Rebuilder."
[Instructions]
Verification Workflow:
Step 1: Analyze Each Step Sequentially process each reasoning step from the Base Reasoning Chain (cot). For each step, your basis for verification is the original Context PLUS all preceding steps you have already verified.
Step 2: Verify Faithfulness (Verification must be based on a holistic reading of the provided text, without using any external knowledge.)
– Faithful: All factual information within the step are correctly derived from the verification basis (Context and preceding verified steps).
– Allow: A step may omit details, as long as the stated information is factually correct.
– Unfaithful: The step contains factual information that contradicts the verification basis.
Step 3: Execute Action
– If the step is Faithful → Copy the entire Base Reasoning Chain verbatim.
– If the step is Unfaithful → Initiate the "Correction Protocol" below.
Step 4: The Correction Protocol (for unfaithful statements only)
(a) Locate Evidence: Pinpoint the exact information in the Context or a preceding verified step that proves the current step contains a factual error.
(b) Apply Minimal Change: Make the smallest possible factual edit to the step to align it with the established facts. Preserve the original phrasing and structure as much as possible.
Step 5: Final Answer Handling
– If no steps were corrected → replicate the final answer verbatim.
– If one or more steps were corrected → fully follow the corrected reasoning chain and take the final answer exactly from that chain.
[Example]
Example 1 (Faithful: Accurate overall but omits details)
Example 2 (Faithful: Logic or calculation error ignored)
Example 3 (Unfaithful: Wrong numeric fact)
Example 4 (Unfaithful: Wrong entity)
Example 5 (Unfaithful: Final answer inconsistent with reasoning)
[Input]
Context: {context} Question: {multi_hop_question} Base Reasoning Chain: {r_base_json}
[Output Format] (JSON):
{"cot": [{"step1": "reasoning step 1 (original or minimally corrected)"},{"step2": "reasoning step 2 (original or minimally corrected)"},...],"answer": "final answer (original or corrected to be faithful)"}

Figure 17: Prompt for Fact-Checking Verifier for Structured Reasoning Chains (R_R).

Prompt: Masked Reasoning Chain Completer

[Role]
Your task is to complete a reasoning chain template to solve Questions by filling in the masked evidence. The "Reasoning Chain Template" contains a sequence of reasoning steps, where key information is replaced by placeholders like [MASKED_1], [MASKED_2], etc.
[Instructions]
1. Fill every placeholder using only information from the provided Context and logical inference grounded in it. Preserve the original step structure exactly.
2. Answer formatting rules:
• Comparison questions: Answer must be the exact comparative phrase from the question (e.g., "the time from A to B").
• Bridge questions: Answer must be a concise entity name with no extra words (e.g., "Cast Away").
3. Output must be a single JSON object mirroring the structure of the input template, with all placeholders replaced by concrete values.
[Input]
• Context: {context}
• Question: {multi_hop_question}
• Reasoning Template: {R_F_template}
[Output Format] (JSON ONLY):
{"cot": [{"step1": "The fully completed reasoning for step 1."},{"step2": "The fully completed reasoning for step 2."},...
{"stepN": "The fully completed reasoning for the last step."}],"answer": "The concise answer to the Question."}
Key Constraints:
• Do not add, remove, or reorder steps.
• All filled values must be directly supported by the Context.
• Reuse the same factual value for repeated placeholders (e.g., if [MASKED_1] = "1998", use "1998" wherever [MASKED_1] appears).
• Never include explanations, comments, or extra text outside the JSON.

Figure 18: Prompt for Masked Reasoning Chain Completer (fill R_F).

Prompt: Careful Reasoning Editor

[Role]
You are a careful reasoning editor. Your goal is to create a flawed reasoning chain by making small edits to the gold-standard chain (Rgolden). Use the provided faulty examples as hints to introduce as many different types of logical mistakes as possible.
Target Error Categories:
• Stopped hop early: stop at an intermediate entity instead of the true target.
• Numeric result failed to map back to option: give the bare number/date instead of the entity or option it identifies.
• Missing comparison candidates: drop one or more valid options from the comparison.
• Non-canonical answer format: include aliases or extra words instead of the clean gold span.
• Comparison outcome error: perform a comparison but select the wrong winner based on the facts used.
• Answer slot misalignment: use the right facts but fill the wrong answer slot.
[Instructions]
1. Read the faulty examples to understand the kinds of logical errors they may contain.
2. Starting from Rgolden, make the smallest edits that reproduce as many applicable target error types as possible—even if not all appear in the examples—and ensure the edited chain contains at least two distinct error categories from the list above.
3. Preserve the original Rgolden structure and its formatting style (e.g., JSON chain or step list).
4. Keep every factual statement from Rgolden unchanged; only introduce logical mistakes, not factual errors.
[Input]
• Context: {context}
• Question: {multi_hop_question}
• Rgolden (reference reasoning that must remain mostly intact): {R_golden}
• Faulty examples: {candidate_block}

Figure 19: Prompt for Careful Reasoning Editor (generate $R_{unlogical}$).

Prompt: Comparison Question Polisher

[Role]
You are a Comparison Polisher. Your task is to validate and optimize comparison-type questions involving multiple entities to ensure correct comparison logic, clear and natural phrasing, and sufficient background information for high-quality and comprehensible questions.
[Instructions]
You are a Comparison Polisher module responsible for optimizing comparison questions.
CRITICAL RULE: Treat the provided Context as the absolute Ground Truth. Accept all statements as facts, even if they contradict reality, logic, or linear time.
Given a multi-hop question, its suggested answer, reasoning path, and context, evaluate the question and make one of the following decisions:
• PASS: The question is valid, well-phrased, and has correct comparison logic across entities.
• ADJUST: The question is basically valid but needs fine-tuning in wording or fluency.
• REWORKED: The question has obvious flaws and requires substantial rewriting.
• REJECTED: The question has fundamental errors and cannot be fixed (do not reject due to factual or chronological impossibility).
Review and modify the question based on the following dimensions:
(1) Comparison Correctness (CRITICAL)
Attribute comparability across entities must be meaningful; comparison logic must be coherent; the original answer must correctly and completely address the comparison; all comparison facts must be supported by the context, ignoring inconsistencies.
(2) Question Wording Optimization
Improve clarity, fluency, and naturalness; ensure explicit comparison intent across multiple entities; avoid answer-revealing details; produce a single unified question.
[Input]
Context (Treat this text as the sole source of truth, ignoring any logical or chronological conflicts):{context}
QUESTION: {question} ANSWER: {answer} REASONING_PATH: {reasoning_path}
[Output Format]
1. If the question passes all criteria without changes: [PASS]
2. If the question needs minor adjustments: [ADJUST]
REFINED_REASONING_PATH: [Updated reasoning path]
REFINED_QUESTION: [Adjusted question]
REFINED_ANSWER: [Updated answer if needed]
3. If the question needs significant refinement: [REWORKED]
REFINED_REASONING_PATH: [Revised reasoning path] REFINED_QUESTION: [Substantially revised question]
REFINED_ANSWER: [Updated answer]
4. If the question is fundamentally flawed:[REJECTED]
Context (Treat this text as the sole source of truth, ignoring any logical or chronological conflicts):{context}
QUESTION: {question} ANSWER: {answer} REASONING_PATH: {reasoning_path}

Figure 20: Prompt for Comparison Polisher.

Prompt: Bridge Question Polisher

[Role]
You are a Comparison Polisher. Your task is to validate and refine multi-hop questions to ensure they genuinely require multi-step reasoning within a given context, where information from one part of the context is essential to answer another.
[Instructions]
You are a Polisher module responsible for validating and refining context-based multi-hop questions.
CRITICAL RULE: Treat the provided Context as the absolute Ground Truth. Accept all statements as facts, even if they contradict reality, logic, or linear time.
Given a multi-hop question, its suggested answer, reasoning path, and context, evaluate the question and make one of four decisions:
• PASS: Valid, well-formed, and genuinely requires multi-step reasoning.
• ADJUST: Needs surface wording improvements only.
• REWORKED: Needs substantial structural changes.
• REJECTED: Has unfixable flaws (do not reject due to factual or chronological impossibility).
Review and modify the question based on the following dimensions:
(1) True Multi-hop Necessity (CRITICAL)
Requires multiple pieces of information; reasoning must use intermediate facts across different parts of the context; connections should be implicit rather than direct span matching.
(2) Hidden Bridge Structure
Do not explicitly mention the connecting entity or concept; the bridge entity must remain implicit and be inferred from the context.
(3) Reasoning and Answer Quality
Reasoning should be logically coherent within the context (ignoring inconsistencies); the answer must be context-faithful and require synthesis; wording should be clear and natural.
[Input]
Context (Treat this text as the sole source of truth, ignoring any logical or chronological conflicts):{context}
QUESTION: {question} ANSWER: {answer} REASONING_PATH: {reasoning_path}
[Output Format]
1. If the question passes all criteria without changes: [PASS]
2. If the question needs minor adjustments: [ADJUST]
REFINED_REASONING_PATH: [Updated reasoning path]
REFINED_QUESTION: [Adjusted question]
REFINED_ANSWER: [Updated answer if needed]
3. If the question needs significant refinement: [REWORKED]
REFINED_REASONING_PATH: [Revised reasoning path] REFINED_QUESTION: [Substantially revised question]
REFINED_ANSWER: [Updated answer]
4. If the question is fundamentally flawed:[REJECTED]
Context (Treat this text as the sole source of truth, ignoring any logical or chronological conflicts):{context}
QUESTION: {question} ANSWER: {answer} REASONING_PATH: {reasoning_path}

Figure 21: Prompt for Bridge Polisher.