Improving Consistency in Retrieval Augmented Systems With Group Similarity Rewards

Faisal Hamman[1,2] **Chenyang Zhu**[1] **Anoop Kumar**[1] **Xujun Peng**[1]
Sanghamitra Dutta[2] **Daben Liu**[1] **Alfy Samuel**[1]
[1]Capital One [2]University of Maryland, College Park

Abstract

Retrieval-Augmented Generation (RAG) systems are increasingly deployed in high-stakes domains where users expect outputs to be consistent across semantically equivalent queries. However, existing systems often exhibit significant inconsistencies due to variability in both the retriever and generator, undermining trust and reliability. In this work, we focus on *information consistency*, i.e., the requirement that generated outputs convey the same core content across semantically equivalent inputs. We introduce a principled evaluation framework that decomposes RAG consistency into retriever-level, generator-level, and end-to-end components, enabling systematic identification of inconsistency sources. To improve consistency, we propose **P**araphrased **S**et Group Relative Policy Optimization (PS-GRPO), an RL approach that leverages multiple rollouts across paraphrased set to assign *group similarity rewards*. We leverage PS-GRPO to achieve Information **Con**sistent **RAG** (Con-RAG), training the generator to produce consistent outputs across paraphrased queries and remain robust to retrieval-induced variability. Because exact reward computation over paraphrase sets is computationally expensive, we also introduce a scalable approximation method that retains effectiveness while enabling efficient, large-scale training. Empirical evaluations across short-form, multi-hop, and long-form QA benchmarks demonstrate that Con-RAG significantly improves consistency while preserving accuracy compared to baselines, even in the absence of explicit ground-truth supervision. Our work provides practical solutions for evaluating and building reliable RAG systems for safety-critical deployments.

1 Introduction

LLMs are increasingly used in open-domain applications where users expect predictable behavior and consistent outputs for semantically equivalent or paraphrased inputs. However, they frequently generate divergent responses to such variations, raising concerns about their reliability and robustness [1–3]. RAG systems are particularly prone to such inconsistencies [4]. RAG architectures combine a retriever and a generator: the retriever selects top-k documents from a large corpus based on the query, and the generator synthesizes a response conditioned on those documents [5]. Semantically similar queries can lead to different retrieved document sets or rankings, resulting in divergent outputs [4]. Weller et al. [6] also highlights a theoretical bottleneck in embedding-based retrieval, showing that the expressivity of top-k retrieval is fundamentally limited; underscoring the need for LLMs that are robust to retrieval inconsistencies. Furthermore, even when the evidence is fixed, the generator may still produce inconsistent responses due to the non-deterministic nature and phrasing sensitivity of LLMs [7].

This inconsistency is particularly problematic in high-stakes domains where RAG systems are commonly deployed, e.g., healthcare, finance, legal [8]. Inconsistent outputs can erode trust, introduce liability risks, or even mislead users [8, 1]. For instance, a customer service RAG assistant may offer

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

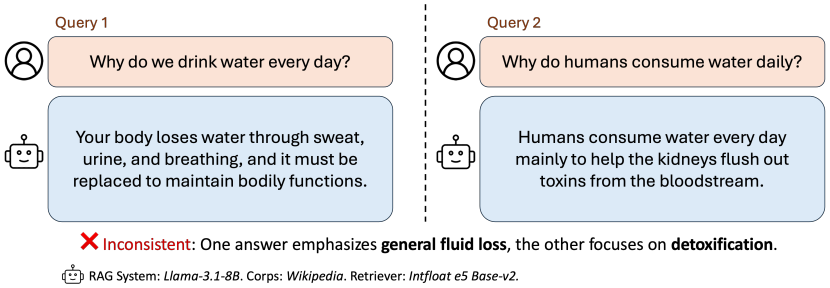

Figure 1: **Motivation.** Semantically equivalent queries lead to different outputs from a RAG system, despite both responses being factually correct. Such variation may be acceptable in many applications, but in certain high-stakes domains (e.g., healthcare, finance, legal) information consistency across equivalent inputs may be required to ensure reliability, user trust, and compliance.

different instructions for "*How do I close my savings account?*" and "*What steps should I take to shut down my savings account?*" despite these queries being semantically equivalent [7].

In this work, we focus on *information consistency*, i.e., the requirement that outputs convey the same core content and information across paraphrased inputs (see motivational Figure 1). The relationship between consistency and accuracy varies across QA tasks. In short-form QA, where there is typically a single concise correct answer, accuracy and consistency are closely linked: a model that answers every paraphrase correctly is necessarily consistent. In contrast, for long-form QA tasks, where multiple valid answers may exist, consistency and accuracy can become orthogonal dimensions: a model can be accurate yet inconsistent, giving different but valid answers across paraphrases, or consistent yet unhelpful, repeating the same wrong or generic answer. Hence, in open-ended tasks, enforcing information consistency becomes a key desideratum alongside answer quality.

Given the practical importance of consistent outputs, we aim to address the following question: *How can we measure and improve the information consistency of RAG system outputs across semantically equivalent inputs, without compromising factual accuracy?* To tackle this, we introduce an evaluation framework that decomposes consistency into retriever-level and generator-level components, and propose a reinforcement learning approach to optimize for consistency using group similarity rewards. Our contributions can be summarized as follows:

- **A Framework for Measuring Consistency in RAG Systems.** We present a principled framework to evaluate consistency in RAG systems by disentangling three components: retriever consistency (Jaccard overlap of documents), generator consistency (LLM outputs given fixed context), and end-to-end consistency. We instantiate it with lexical (e.g., BLEU) and LLM-as-a-judge similarity, which lets us attribute inconsistency to the retriever or the generator, quantify how retrieval variation propagates to the final output (see Section 2.1).
- **Con-RAG: Improving Consistency via Paraphrased Set GRPO.** To improve consistency across semantically equivalent queries, we propose **P**araphrased **S**et GRPO, an RL approach that leverages multiple rollouts across a set of paraphrased inputs to assign *group similarity rewards*. This forms the core of our Information **Con**sistent **RAG** (Con-RAG) framework (see Figure 3). Due to complexity of computing the rewards, we introduce a relaxed approximation by subsampling paraphrases and rollouts, reducing the number of comparisons from quadratic to linear in the number of paraphrases. This allows us to train Con-RAG efficiently on large datasets while preserving reward fidelity (see Section 2.2).
- **Empirical Evaluation.** We conduct an extensive evaluation of Con-RAG across six QA benchmarks: short-form, multi-hop, and long-form QA tasks on Llama-3.1 and Qwen-2.5 model families[1] (see Figure 2). Our results show that Con-RAG significantly improves both end-to-end and generator consistency compared to baselines, while improving or preserving accuracy. This also holds for long-form QA tasks trained without ground-truth supervision.

[1]Code: `https://github.com/CapitalOne-Research/con-rag`

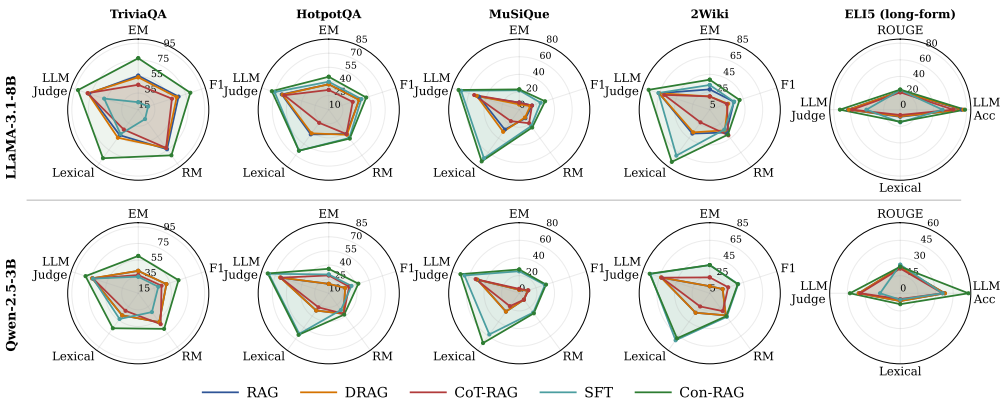

Figure 2: **Comparison between Con-RAG and baselines across accuracy and consistency dimensions on LLaMA-3.1-8B and Qwen-2.5-3B.** Each plot summarizes performance on a single dataset using accuracy measures (Exact Match, Token F1, Relaxed Match) and end-to-end information consistency (measured lexically (BLEU) and LLM-as-a-judge). Con-RAG outperforms prior methods across models, achieving higher factual accuracy and more consistent responses across paraphrased inputs (see Table 2 & 3 for full results including long form Natural Questions dataset).

1.1 Related Work

Consistency in Language Models. Consistency has emerged as a key concern for safety and reliability in high-stakes LLM deployment [8, 1]. Prior work has introduced various notions of consistency. Logical consistency refers to the ability of the model to make decisions without logical contradiction [9–12]. Factual consistency, often discussed as faithfulness or hallucination, considers whether model outputs contradict the source content [9, 13–15]. Self-consistency evaluates whether similar inputs yield stable explanations [16]. Nonlogical forms of consistency, such as moral consistency, assess coherence of values across contexts [17, 18]. Prediction consistency examines whether a model's predictions remain stable across multiple fine-tuned variants that achieve similar overall performance [19, 20]. Closest to our work is semantic consistency, which measures output stability under semantically equivalent inputs like paraphrases. This has been evaluated using datasets like ParaRel [2] and metrics such as BERTScore, entailment scores, and LLM judges [21–23]. Approaches to improve semantic consistency include custom losses [2], knowledge distillation from consistent teachers [3], and synthetic data supervision [24]. We refer to a recent survey exploring current landscape and future directions in consistency in LLMs [1].

Consistency in RAG Systems. RAG improves factual accuracy by conditioning outputs on retrieved evidence [25–27]. However, it introduces new sources of inconsistency due to retriever sensitivity and generator (LLM) variability. Despite growing use in high-stakes applications, information consistency in RAG remains underexplored, with the exception of a few notable studies addressing robustness in retrieval or prompt-level variation [28–30, 4]. Orthogonal work improves RAG accuracy, rather than consistency, through retrieval-aware training or adaptive retrieval [31–35]. Our work aims to evaluate and improve information consistency in RAG, leveraging an RL-based optimization with group similarity rewards. Our approach builds on recent advances in RL for LLMs [36], particularly GRPO [37], which trains on verifiable reward assignment across outputs. Our framework improves information consistency across semantically equivalent inputs without requiring ground-truth labels, unlike prior consistency-improvement methods [3, 24]. Recent label-free post-training methods also learn from model-derived signals, such as self-certainty, majority-vote pseudo-labels, self-generated targets, or sharpened base distributions [38–41]; these rewards are defined on a single input, whereas PS-GRPO's reward is relational across paraphrased inputs.

2 Main Contributions

In this section, we first define a framework to measure consistency in RAG systems by isolating retriever, generator, and end-to-end contributions (see Section 2.1), then introduce our Con-RAG method to improve consistency via group similarity rewards and its relaxation (see Section 2.2).

2.1 Measuring Consistency in RAG Systems

We consider a RAG system composed of a retriever R and a generator (LLM). Given a user query q, the system first retrieves a set of top-k documents from a corpus $\mathcal{D}$, and then generates an output $y = \text{LLM}(q, R(q))$ conditioned on these documents:$R(q) = \{d_1, \ldots, d_k\} \subset \mathcal{D}$. Let q_0 be a canonical input query, and let $\mathcal{P}(q_0) = \{p_1, p_2, \ldots, p_n\}$ denote a set of paraphrased or semantically equivalent inputs. Our goal is to assess the *output consistency* of the RAG system across this paraphrased set.

Retriever Consistency. Retriever consistency measures how stable retrieval is across paraphrased queries. Let $R(p_i)$ be the set of documents retrieved for p_i; for each pair (p_i, p_j), we compare $R(p_i)$ and $R(p_j)$ using *Jaccard similarity* [42], the ratio of the size of their intersection to the size of their union. Jaccard similarity equals 1 when both paraphrases retrieve exactly the same documents and 0 when they share none, and normalizing by the union makes it comparable across retrieval depths. We define retriever consistency as the average Jaccard similarity over all unique paraphrase pairs:

$$\mathcal{C}_{\text{ret}}(q_0) = \frac{2}{n(n-1)} \sum_{i<j} \frac{|R(p_i) \cap R(p_j)|}{|R(p_i) \cup R(p_j)|}.$$

End-to-End RAG Consistency. Let $y_i = \text{LLM}(p_i, R(p_i))$ denote the output of the RAG system for paraphrase p_i. End-to-end consistency measures alignment across outputs when the entire pipeline is allowed to vary, each paraphrase p_i is passed to the retriever, which may return a different document set $R(p_i)$, and the generator then conditions on this evidence to produce y_i. Formally, we compute pairwise similarity across all outputs:

$$\mathcal{C}_{\text{e2e}}(q_0) = \frac{1}{n(n-1)} \sum_{i \neq j} \text{sim}(y_i, y_j).$$

This captures the overall stability of the RAG system under paraphrased inputs, reflecting the combined variability introduced by both retrieval and generation. The similarity function $\text{sim}(\cdot, \cdot)$ can be instantiated with lexical metrics (e.g., BLEU, ROUGE), embedding-based similarity (e.g., BERTScore), entailment scores from NLI models, or an LLM-as-a-judge that assesses whether y_i and y_j convey the same information.

Generator (LLM) Consistency. To isolate the generator's contribution, we can fix the retrieved documents across all paraphrases and measure similarity among the outputs, i.e., $y_i^{\text{fixed}} = \text{LLM}(p_i, R(q_0))$, and compute consistency over $\{y_1^{\text{fixed}}, \ldots, y_n^{\text{fixed}}\}$. This captures how consistently the LLM alone responds to semantically equivalent inputs when conditioned on identical evidence. Conceptually, this is closely related to prior work on consistency in standalone LLMs, where the focus is on ensuring paraphrase-invariant outputs under similar prompts [2, 3, 1, 7].

2.2 Improving Consistency via Paraphrased Set GRPO

Given a RAG system comprising a retriever R and generator (LLM), and a canonical query q_0 with paraphrases $\mathcal{P}(q_0) = \{p_1, \ldots, p_n\}$, our goal is to maximize output consistency while preserving, or improving factual accuracy. We propose Paraphrased Set GRPO, an RL algorithm that leverages GRPO's multiple rollouts across paraphrased inputs to assign group similarity rewards. Our objective is to directly train the generator so that outputs across semantically equivalent inputs are consistent.

Group Relative Policy Optimization. GRPO is an RL optimization algorithm that estimates advantage through group-normalized rewards rather than using a critic model [37]. For a given query q, GRPO samples a group of g rollouts, i.e., multiple possible completions generated from the policy under stochastic decoding (such as temperature or nucleus sampling), denoted by $\{o_1, \ldots, o_g\}$, where each rollout is drawn as $o_i \sim \pi_\theta(\cdot \mid q)$. Each rollout receives a verifiable scalar reward $r_i = \text{Reward}(o_i \mid q)$, and the normalized advantage for each rollout is computed as $\hat{A}_i = (r_i - \mu_q)/\sigma_q$, where μ_q and σ_q are the mean and standard deviation of rewards within the group. Let $y_{i,1:|o_i|}$ denote the tokens of response o_i, and let $\rho_{i,t} = \pi_\theta(y_{i,t} \mid q, y_{i,<t}) \,/\, \pi_{\theta_{\text{old}}}(y_{i,t} \mid q, y_{i,<t})$ be the token-level importance ratio. The policy is then optimized by maximizing the objective using these group-relative

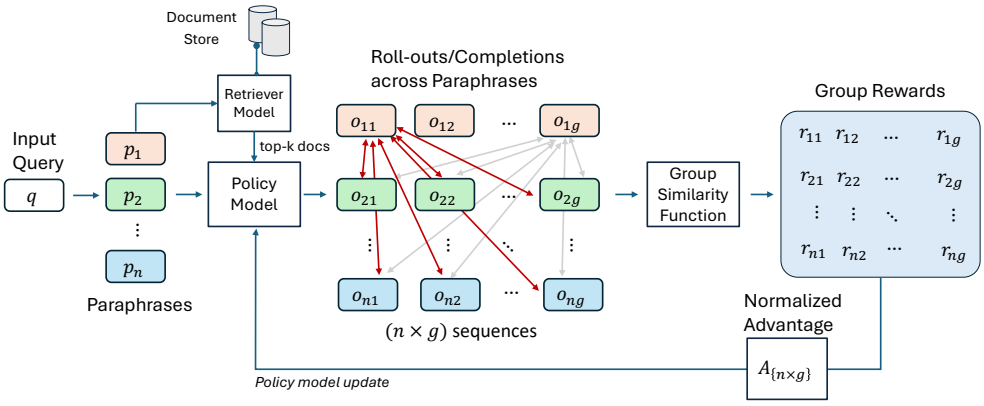

Figure 3: Overview of PS-GRPO and Information **Con**sistent **RAG** (Con-RAG) framework. A canonical query q is expanded into a set of paraphrases $\{p_1, \ldots, p_n\}$, each of which is passed through the policy model to generate g sampled rollouts. For every rollout o_{ij}, we compute a group similarity reward r_{ij} by averaging its similarity with outputs from other paraphrases of the same query (this produces an $n \times g$ reward matrix). Normalized advantages are then computed within each paraphrase set, and the policy model is updated.

advantages, with an optional KL penalty to penalize deviation from the reference policy:

$$\mathcal{L}(\theta) = \frac{1}{g} \sum_{i=1}^{g} \sum_{t=1}^{|o_i|} \min\Big(\rho_{i,t}\hat{A}_i,\ \text{clip}(\rho_{i,t},\ 1-\epsilon,\ 1+\epsilon)\hat{A}_i\Big) - \beta\, \mathbb{D}_{\text{KL}}\big(\pi_\theta(\cdot \mid q) \,\big\|\, \pi_{\text{ref}}(\cdot \mid q)\big)$$

PS-GRPO with Group Similarity Rewards. PS-GRPO introduces a group-level objective that promotes consistent generation across semantically equivalent queries. It leverages the unique property of GRPO which generates *extensive rollouts per query*. We aggregate all rollouts from all paraphrases into a group and compute similarity-based rewards *across the paraphrase set*, so each output is rewarded according to its similarity with outputs generated for the other paraphrases of the same canonical query. For each canonical query q_0 with paraphrases $\mathcal{P}(q_0) = \{p_1, \ldots, p_n\}$, the policy LLM π_θ generates g rollouts per paraphrase: $o_{ij} \sim \pi_\theta(\cdot \mid p_i, R(p_i)), i \in \{1, \ldots, n\},\ j \in \{1, \ldots, g\}$. Collect these into an $n \times g$ matrix $\{o_{ij}\}$ (total $n \times g$ rollouts). We assign each rollout o_{ij} a group similarity reward by averaging its similarity to all rollouts generated for the *other* paraphrases (also see Figure 3 for illustration):

$$r_{ij} \;=\; \frac{1}{(n-1)g} \sum_{\substack{u=1 \\ u \neq i}}^{n} \sum_{m=1}^{g} \text{sim}\big(o_{ij},\, o_{um}\big), \tag{1}$$

where $\text{sim}(\cdot, \cdot)$ is the similarity function. In practice, we instantiate sim using the BLEU metric, motivated by recent findings that BLEU serves as a strong proxy for reward models in aligning LLMs with human preferences [43]. As further validated in our ablation study (see Table 4), lower-order BLEU outperformed alternative lexical similarity metrics, and matched BERTScore reward on information consistency at substantially lower compute cost (see Appendix A.10). Group-normalized advantages are then computed across each paraphrased rollout:$\hat{A}_{ij} = (r_{ij} - \mu_i)/\sigma_i$, with μ_i, σ_i the mean and standard deviation of rewards for rollouts for p_i. The policy is optimized with the standard GRPO clipped objective using $\hat{A}_{ij}$ and (optionally) a KL penalty to a reference policy with weight β. If ground-truth answers are available (e.g., in short-form QA tasks), we extend the reward to improve consistency and accuracy. Specifically, for each rollout we define a combined weighted reward:

$$r_{ij}^{\text{final}} \;=\; \alpha\, r_{ij}^{\text{cons}} \;+\; \gamma\, \text{Acc}(o_{ij}, y^\star), \tag{2}$$

where r_{ij}^{cons} is the group similarity reward, $y^\star$ is the ground-truth answer, and $\text{Acc}(\cdot, \cdot)$ is measured using token F1 score. Importantly, our method does not require ground truths to improve consistency: the accuracy reward term can be omitted, as demonstrated in our long-form QA experiments (see Section 3), where questions are open-ended and no single ground-truth answer exists.

Efficient Computation of Group Similarity Rewards for Scalable Training. Computing group similarity rewards can be expensive, especially in a training environment where rewards must be

Table 1: **Disentangling sources of inconsistency in RAG systems (LLaMA-3.1-8B)**. Retriever consistency is low across datasets, suggesting that paraphrased queries often retrieve non-overlapping documents. This introduces context variability that is reflected in the end-to-end consistency scores. Fixing retrieval improves consistency, but variation remains, revealing the generator's sensitivity to input phrasing even with identical evidence. We present accuracy values in Table 7 (also see Table 8 for Qwen-2.5-3B).

Dataset	End-to-End Consistency		Generator (LLM) Consistency		Retriever Consistency
	Lexical	LLM-Judge	Lexical	LLM-Judge	Jaccard Overlap
TriviaQA	53.0	77.8	67.3	88.5	32.5
HotpotQA	42.5	62.5	53.7	71.9	46.0
2Wiki	38.5	65.5	48.4	76.4	52.4
MuSiQue	27.9	48.2	44.4	69.7	36.6
ElI5	8.6	62.8	15.1	74.2	27.1
NQ	14.2	53.2	22.4	68.2	38.9

computed at every gradient step. For each rollout o_{ij}, computing its reward requires comparing against all $(n-1)g$ rollouts from the other paraphrases. At the query level, with n paraphrases and g rollouts each, the naive total cost is $ng \times (n-1)g \;=\; n(n-1)g^2$ similarity computations. For example, with $n = 5$ and $g = 6$ amounts to 720 comparisons for a single query. Exploiting symmetry reduces this to $\frac{1}{2}\, n(n-1)g^2$, but the cost still scales quadratically with n and g. To make training feasible, we introduce a *relaxed group similarity reward*. Instead of averaging over all cross-paraphrase comparisons, for each rollout o_{ij} we uniformly subsample κ paraphrases $K \subset \{1, \ldots, n\} \setminus \{i\}$ and s rollouts per chosen paraphrase $S_u \subset \{1, \ldots, g\}$, and approximate:

$$\tilde{r}_{ij} \;=\; \frac{1}{\kappa s} \sum_{u \in K} \sum_{m \in S_u} \mathrm{sim}(o_{ij},\, o_{um}),$$

which is an unbiased estimator under uniform sampling. This reduces the per-query cost from $O(n(n-1)g^2)$ to $O(ng\kappa s)$, if $\kappa \ll n-1$ and $s \ll g$. In practice, this approximation preserves the training signal while keeping the reward computation tractable. We conduct ablations to study the impact of this subsampling on both performance and computational cost across a wide range of (κ, s) settings (Appendix A.9).

3 Experiments

In this section, we describe our experimental setup to evaluate Con-RAG on various QA tasks, describing datasets, paraphrase generation, consistency metrics, training details, and baselines.[2]

Datasets. We evaluate our approach across three types of question answering (QA) tasks: Short-form QA tasks: TriviaQA [44] and HotpotQA [45], both requiring concise fact-based answers. Multi-hop QA tasks: 2WikiMultiHopQA [46] and MuSiQue [47], which involve reasoning over multiple pieces of evidence. Long-form QA tasks: ELI5 [48] and Natural Questions (NQ) [49], where answers are open-ended and typically span multiple sentences. None of these datasets provide paraphrased versions of the input questions. Hence, we synthetically generate paraphrases for each query using LLaMA-3.1-70B. Specifically, for each original query q_0, we generate n paraphrases $\mathcal{P}(q_0) = \{p_1, \ldots, p_n\}$, enabling us to simulate semantically equivalent inputs (see Appendix A.2 for prompts and details).

Setup. Our RAG system consists of a LLaMA-3.1-8B and Qwen-2.5-3B model serving as the generator, and a dense retriever built on top of the intfloat/e5-base-v2 embedding model [50]. We use KILT Wikipedia snapshot [51] as our document corpus, where each article is segmented into chunks of 512 tokens. At inference time, the retriever selects the top-$k = 5$ documents per query, which are then appended to the prompt for generation. To isolate effects from sampling inconsistencies, we use deterministic decoding for all evaluations.

Evaluating Consistency in RAG Systems. We evaluate performance along two dimensions: accuracy and consistency. For short-form and multi-hop QA datasets, accuracy is measured using: (i) Exact Match (EM), (ii) token F1 score, and (iii) Relaxed Match (RM), which considers an answer correct if

[2]Code: `https://github.com/CapitalOne-Research/con-rag`

Table 2: **Comparison between Con-RAG and Baselines** (Short-form QA, LLaMA-3.1-8B). Lexical consistency measured via BLEU score while information consistency measured using an LLM-as-a-judge. Con-RAG is trained with a group similarity reward plus an accuracy reward (no KL), and consistently yields higher end-to-end and generator-only consistency while also improving accuracy over original queries (see radar plot illustration in Figure 2). Refer to Table 10 for results on Qwen-2.5-3B model.

		Accuracy (%)			**End-to-End Consistency (%)**		**Generator (LLM) Consistency (%)**	
Dataset	**Method**	EM	F1	RM	Lexical	Inform.	Lexical	Inform.
TriviaQA	RAG	56.0	66.1	74.0	53.0	77.8	67.3	88.5
	DRAG	54.0	63.7	72.0	56.8	78.7	68.2	88.2
	CoT-RAG	45.0	57.7	72.0	44.6	79.2	57.7	85.0
	SFT	24.0	27.5	29.0	51.3	58.2	77.8	81.2
	Con-RAG	**77.0**	**81.0**	**83.0**	**87.3**	**91.3**	**91.2**	**93.0**
HotpotQA	RAG	37.0	44.1	42.0	42.5	62.5	53.7	71.9
	DRAG	37.0	43.8	43.0	41.1	61.6	50.5	73.1
	CoT-RAG	31.0	36.8	42.0	27.3	59.6	36.1	68.9
	SFT	39.7	46.5	47.2	**63.9**	70.5	72.2	78.5
	Con-RAG	**45.0**	**51.9**	**48.0**	**63.9**	**73.6**	**80.9**	**88.2**
MuSiQue	RAG	8.0	15.3	12.0	27.9	48.2	44.4	69.7
	DRAG	6.0	13.1	11.0	31.0	50.7	42.9	70.0
	CoT-RAG	8.0	15.2	19.0	16.1	53.7	29.2	67.7
	SFT	22.0	25.5	23.0	68.1	69.3	77.8	79.8
	Con-RAG	**23.0**	**30.8**	**25.0**	**72.5**	**72.3**	**91.4**	**92.7**
2Wiki	RAG	28.0	33.9	37.0	38.5	65.5	48.4	76.4
	DRAG	20.0	26.9	34.0	36.8	65.5	49.3	76.1
	CoT-RAG	20.0	25.5	41.0	22.8	59.3	29.9	67.8
	SFT	33.0	34.0	33.0	69.4	66.2	84.4	83.3
	Con-RAG	**39.0**	**40.6**	**40.0**	**78.2**	**77.8**	**94.1**	**95.5**

the ground truth answer appears anywhere in the output. For long-form QA (e.g., ELI5, NQ), where answers are open-ended and may be phrased in diverse ways, EM/F1/RM are too restrictive. Instead, we evaluate accuracy using: (i) ROUGE, to capture content overlap with reference answers, and (ii) an LLM-as-a-judge, in which a strong model (LLaMA-3.3-70B) assesses whether the generated answer is factually correct; we report this as LLM-Acc. Consistency is evaluated at three levels (disentangling contributions from the retriever and generator): (i) End-to-end consistency, where each paraphrase retrieves its own documents and we compute agreement between outputs (via BLEU for lexical consistency and an LLM-as-a-judge for information consistency; see Appendix A.3 for prompts); (ii) Generator consistency, where retrieval is fixed across paraphrases and agreements across outputs are measured; (iii) Retriever consistency, defined as the average Jaccard overlap between retrieved document sets across paraphrases (see Section 2.1). We use paraphrase size $n = 5$ for evaluations.

We summarize consistency results across the datasets in Table 1. We observe that the retriever consistency is relatively low across the datasets, indicating that paraphrases often retrieve non-overlapping sets of documents. This is reflected in the end-to-end consistency scores, which show that these small changes in query phrasing can result in different answers, due to shifts in both retrieved context and model generation. To isolate the generator's contribution, we also evaluate generator consistency under fixed retrieval (i.e., same documents across paraphrases). While consistency scores improve, substantial variability still remain, showing that even with identical evidence, the generator (LLM) exhibits sensitivity to input paraphrasing. We further quantify how retrieval variation propagates into end-to-end inconsistency through a per-query analysis in Table 12 in Appendix A.6). We report accuracy for original queries, paraphrased queries, and paraphrased queries with fixed documents in Table 7. Across these settings, accuracy remains relatively stable, with only minor fluctuations, suggesting that paraphrasing and retrieval shifts have limited impact on final answer correctness on average.

Con-RAG Training Setup. We train Con-RAG with BLEU as similarity function for computing group similarity rewards. For short-form and multi-hop QA tasks, we use unigram BLEU-1 and bigram BLEU-2 for long-form QA tasks to account for more contextual similarity across longer

Table 3: **Comparison between Con-RAG and Baselines** (Long-form QA Task). Con-RAG is trained using only the group similarity rewards with a small KL regularizer (no accuracy supervision). Despite no ground-truth, it achieves the best information consistency and also maintains answer quality over baselines, whereas SFT on reference answers underperforms in this open-ended setting.

		Accuracy (%)		**End-to-End Consistency (%)**		**Generator (LLM) Consistency (%)**	
Dataset	**Method**	ROUGE	LLM-Acc	Lexical	Inform.	Lexical	Inform.
ELI5	RAG	21.9	74.0	8.6	62.8	15.1	74.2
	DRAG	22.0	76.0	8.0	62.2	15.0	72.5
	CoT-RAG	20.9	64.0	6.4	57.8	10.3	71.0
	SFT	23.5	51.0	**15.3**	40.8	16.6	41.7
	CoG	23.1	55.3	15.2	45.7	17.2	48.4
	CAT	22.8	58.1	14.7	49.0	16.3	50.9
	Con-RAG	**24.2**	**78.0**	14.6	**72.7**	**21.7**	**80.8**
NQ	RAG	26.9	**61.0**	14.2	53.2	22.4	68.2
	DRAG	26.9	59.0	14.2	53.5	22.4	68.2
	CoT-RAG	24.3	55.0	12.1	54.7	19.8	70.5
	SFT	**36.8**	22.0	**37.1**	36.2	**56.2**	55.0
	CoG	28.4	48.0	24.5	46.8	38.2	63.5
	CAT	27.1	53.0	20.8	52.1	33.5	66.8
	Con-RAG	23.5	59.0	22.8	**59.3**	32.3	**75.5**

answers (see Appendix A.4 for background on BLEU-n). For short-form QA tasks, where ground-truth answers are available, we augment the similarity reward with an accuracy reward based on token F1 score, which we found to be more stable than other accuracy metrics. The final reward is computed using a weighted sum as defined in Eq. 2, with equal weights $(\alpha, \gamma = 1)$ for both consistency and accuracy. We set the KL regularization coefficient β=0 for these tasks, following recent findings [52] suggesting that GRPO performs effectively without explicit KL penalties. In contrast, for long-form QA (ELI5, NQ), where questions are open-ended and multiple valid answers may exist, we exclude the accuracy reward and optimize solely for consistency using the group similarity rewards. To prevent reward hacking in the absence of ground-truth, we apply a small KL penalty with β=0.05 to regularize the policy against a reference model (see Appendix A.13 for a discussion and guidance on choosing β).

We use $n = 6$ paraphrases per canonical query and $g = 4$ rollouts per paraphrase. To make training scalable, we apply the relaxed approximation described in Section 2.2 to estimate group similarity rewards. Specifically, we subsample $\kappa = 3$ paraphrases and $s = 1$ rollout per selected paraphrase when computing similarity, which significantly reduces the number of comparisons with minimal impact on reward quality. We perform full model fine-tuning using the AdamW optimizer with a learning rate of `1e-6`. All training is conducted on LLaMA-3.1-8B and Qwen-2.5-3B. Full training configurations details are provided in Appendix A.5.

Baselines. All baselines share the same retriever, generator, and corpus. *Prompting:* (i) **RAG**, which appends the top-k retrieved documents to the prompt; (ii) **DRAG** (Demonstrated RAG) [53], which adds few-shot demonstrations; (iii) **CoT-RAG** (Chain-of-Thought RAG) [54], which elicits intermediate reasoning before the answer. *Training-based:* (iv) **SFT** (Supervised Fine-Tuning) [55] on paraphrased queries paired with ground-truth answers (reference responses for long-form QA) *Consistency alignment:* (v) **CoG** (Chain of Guidance) [3], which fine-tunes the generator on paraphrase-consistent answers selected by an LLM ranking pipeline; (vi) **CAT** (Consistency Alignment Training) [24], which follows paraphrase-augmented SFT with a self-rewarded ranking loss that prefers responses judged correct against the reference. Both were proposed for standalone LLMs; we adapt them to RAG by conditioning on retrieved documents. We evaluate CoG and CAT on long-form QA, where Con-RAG is trained without ground truth, making the comparison against these reference-supervised consistency methods most informative.

Results and Analysis. We present our results across short-form and long-form QA tasks in Figure 2 and Tables 2. To show that consistency improvements do not come at the cost of answer quality, we report accuracy metrics on the original queries, avoiding generic but consistent outputs. Our results demonstrate the following key observations:

Con-RAG improves both consistency and accuracy in short-form QA. Across all short and multi-hop datasets, Con-RAG achieves significant gains in both end-to-end and generator-only consistency. For instance, on TriviaQA, end-to-end consistency (lexical/information) improves from $53.0/77.8$ (RAG) to $87.3/91.3$, while generator consistency reaches $91.2/93.0$. Notably, improvements are not achieved at the expense of accuracy. Con-RAG also achieves the highest EM, F1, and RM scores across all datasets (see Table 2). This indicates that optimizing consistency can also enhance model robustness, likely due to the implicit data augmentation effect of training across paraphrased inputs (see Appendix A.11). DRAG and CoT-RAG provide only modest consistency improvements and fail to match Con-RAG.

Table 4: **Effect of reward similarity metric on Con-RAG** (ELI5, Qwen-2.5-3B). Lower-order BLEU better preserves semantic adequacy across paraphrases, while stricter metrics (higher-order BLEU, EM) penalize valid rephrasings. BLEU-2 yields the best consistency and accuracy.

Metric	Acc		E2E Cons		Gen Cons	
	Rouge	LLM	Lex	Inf	Lex	Inf
BLEU-1	**22.6**	54.0	6.9	38.2	14.8	**69.8**
BLEU-2	22.5	**58.0**	**9.2**	**42.0**	**17.8**	67.5
BLEU-3	22.4	49.0	6.7	36.3	14.8	66.0
BLEU-4	22.2	50.0	6.4	36.2	14.2	66.5
ROUGE-L	22.1	46.0	6.1	35.2	13.6	65.2
Exact Match	22.1	49.0	6.6	37.7	14.4	66.2

In Long-form QA, Con-RAG improves consistency even without ground-truth supervision. On ELI5, despite being trained without any explicit ground-truth, Con-RAG improves both consistency and accuracy over baselines. Compared to RAG, Con-RAG increases lexical and information consistency while also achieving higher ROUGE and LLM-judged accuracy (see Table 3). For NQ dataset, consistency improves without taking a hit in accuracy. In contrast, SFT trained on reference answers performs poorly, especially in terms of LLM-judge accuracy, highlighting the limitations of rigid supervision in open-ended QA. We further show that these gains transfer out of distribution: Con-RAG trained on one long-form dataset and evaluated on the other preserves accuracy close to the RAG baseline while still improving consistency (see Appendix A.12).

Ablation Studies. To analyze design choices in Con-RAG, we run focused ablations on a lighter generator, Qwen-2.5-3B for fast, controlled sweeps.

(1) *Varying similarity function used in the group reward.* We replace BLEU in the group similarity reward with alternative choices and measure resulting consistency/accuracy. We consider: BLEU-n ($n \in \{1, 2, 3, 4\}$), ROUGE-L, Exact Match (results are summarized in Table 4). Lower order BLEU yields the best consistency and accuracy. We further replace BLEU with BERTScore, a semantic similarity reward, which achieves comparable LLM-Judge consistency and accuracy with lower lexical consistency, but at higher computational cost (see Appendix A.10).

(2) *Subsampling (κ, s) sensitivity and compute trade-offs.* We analyze the sensitivity of Con-RAG to the subsampling parameters (κ, s) used in the relaxed group-similarity reward, showing stable performance across a wide range of settings with increasing compute costs at larger budgets. Detailed results and discussion are provided in Appendix A.9.

(3) *Comparison with standard GRPO.* We compare PS-GRPO against standard GRPO trained with a pointwise BLEU reward to ground-truth answers. PS-GRPO consistently achieves higher accuracy and consistency, despite using no ground-truth supervision. The learning signal from mutual agreement across paraphrases is more effective than simply pushing each response toward a single reference. Full results and discussion are provided in Appendix A.7.

(4) *Varying short-form accuracy reward metrics.* On short-form QA, we study the effect of different reward signals on accuracy and consistency by conducting ablations with: (i) consistency term only training, (ii) accuracy term only training, and (iii) joint training with consistency plus accuracy. For the accuracy component, we compare token F1 (ours), EM, and RM (see Table 5). Best performance is achieved when combining consistency with the token F1 reward.

(5) *Robustness to decoding strategy and retrieval depth.* We evaluate how inference-time stochasticity impacts consistency and accuracy by sweeping the decoding temperature $T \in \{0.0, 0.5, 1.0, 2.0\}$ (Table 19) and by evaluating Con-RAG under greedy, top-p, top-k, and temperature sampling. We also vary the number of retrieved documents ($k \in \{5, 10, 15\}$). Con-RAG maintains consistency gains over the RAG baseline across all decoding strategies and retrieval depths (see Appendix A.14).

Table 5: **Effect of Accuracy Reward Variant on Con-RAG** (TriviaQA, Qwen-2.5-3B). We compare consistency-only training, accuracy-only training, and joint training with consistency plus various accuracy metrics. The best performance is achieved when combining consistency with the token F1 reward, which yields the highest accuracy and consistency values.

Reward Variant	α	γ	**Accuracy (%)**			**End-to-End Cons. (%)**		**Generator Cons. (%)**	
			EM	F1	RM	Lexical	Inform.	Lexical	Inform.
Consistency only	1.0	0.0	51.5	53.2	59.0	59.9	79.0	78.7	88.0
Accuracy only (F1)	0.0	1.0	54.0	56.0	60.4	52.0	75.0	62.0	84.1
Consistency + EM	1.0	1.0	56.2	63.5	65.0	61.5	80.2	76.0	88.4
Consistency + RM	1.0	1.0	57.0	64.0	66.0	62.3	80.5	77.0	88.5
Consistency + F1	1.0	1.0	**60.0**	**66.0**	**68.0**	**67.1**	**81.8**	**80.5**	**89.5**

4 Discussion

BLEU metric as a proxy reward signal. BLEU metrics alone do not capture semantic consistency. The metric serves as a lightweight proxy signal that nudges the model towards more consistent outputs across paraphrases. Our use of BLEU is also driven by scalability considerations; group-level reward computation must be performed hundreds of times per optimization step, making an efficient metric essential. Moreover, prior work [43] shows that lower-order BLEU correlates surprisingly well with human preference signals (also see ablations in Table 4). Importantly, even under such minimal reward, Con-RAG improves not only lexical overlap but also LLM-judge consistency, which directly evaluates information-level agreement. Stronger semantic rewards can be substituted, but at increasing cost: replacing BLEU with BERTScore yields comparable information consistency and accuracy at substantially higher compute (Appendix A.10), while an LLM-based consistency critic would be prohibitively expensive at this scale. In this work, we demonstrate that (1) group similarity rewards even without ground truths is an effective framework for optimizing consistency, and (2) even with a simple and efficient similarity function, the method produces measurable gains in information consistency.

Beyond Lexical Rewards for Information Consistency. An important direction for future work is to develop reward signals that directly optimize information consistency without relying on lexical similarity. While LLM-as-a-judge metrics are promising, they introduce a trade-off between weak and strong supervision [56]. Our BERTScore experiment (Appendix A.10) shows that semantic rewards preserve information consistency while tolerating lexical diversity, but at a higher compute cost. Ideally, we seek lightweight, efficient signals that can guide models toward consistent information content, such as entailment-based rewards or cheaper embedding-based approaches.

Retrieval-induced variability. Our approach is retriever-agnostic and can be applied on top of any retrieval system. We show that even when the retriever is perfectly consistent, i.e., when we freeze the retrieved documents across paraphrases, substantial inconsistency remains at the generator level. This motivates our focus on improving the generator to remain robust to retrieval-induced variability. This design choice becomes more important as LLMs move toward agentic systems that interleave reasoning with calls to external tools such as web search [57–59]. In these settings, retrieval is often provided by third-party, API-based search services that the system builder cannot modify or train, so variability in what is retrieved must be absorbed by the model itself. Methods that train only the generator, like ours, therefore remain applicable even when the retrieval component is a black box. We also note that our approach is not limited to RAG: it applies broadly to standalone LLMs, where paraphrase sensitivity and generation inconsistency persist even without retrieval. We focus on RAG because the variability introduced by the retriever amplifies these inconsistencies and makes the problem more critical in practice. When the retriever is accessible, jointly training or regularizing the retriever and generator is a promising direction for future work [27, 31, 32]. By introducing a principled way to measure consistency and a scalable method to improve it, we take a step towards more reliable and user-aligned RAG systems. We discuss broader impacts in Appendix A.1.

Acknowledgment. Part of this work was done during an internship at Capital One and was funded by Capital One. Additional support was provided by the Ann G. Wylie Dissertation Fellowship from the University of Maryland Graduate School.

References

[1] Jekaterina Novikova, Carol Anderson, Borhane Blili-Hamelin, Domenic Rosati, and Subhabrata Majumdar. Consistency in language models: Current landscape, challenges, and future directions. *arXiv preprint arXiv:2505.00268*, 2025.

[2] Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, 2021. doi: 10.1162/tacl_a_00410. URL `https://aclanthology.org/2021.tacl-1.60/`.

[3] Harsh Raj, Vipul Gupta, Domenic Rosati, and Subhabrata Majumdar. Improving consistency in large language models through chain of guidance. *arXiv preprint arXiv:2502.15924*, 2025.

[4] Sezen Perçin, Xin Su, Qutub Sha Syed, Phillip Howard, Aleksei Kuvshinov, Leo Schwinn, and Kay-Ulrich Scholl. Investigating the robustness of retrieval-augmented generation at the query level. *arXiv preprint arXiv:2507.06956*, 2025.

[5] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1), 2023.

[6] Orion Weller, Michael Boratko, Iftekhar Naim, and Jinhyuk Lee. On the theoretical limitations of embedding-based retrieval. *arXiv preprint arXiv:2508.21038*, 2025.

[7] Amirhossein Razavi, Mina Soltangheis, Negar Arabzadeh, Sara Salamat, Morteza Zihayat, and Ebrahim Bagheri. Benchmarking prompt sensitivity in large language models. In *European Conference on Information Retrieval*, pages 303–313. Springer, 2025.

[8] Sunnie SY Kim, Jennifer Wortman Vaughan, Q Vera Liao, Tania Lombrozo, and Olga Russakovsky. Fostering appropriate reliance on large language models: The role of explanations, sources, and inconsistencies. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2025.

[9] Myeongjun Jang, Deuk Sin Kwon, and Thomas Lukasiewicz. Becel: Benchmark for consistency evaluation of language models. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3680–3696, 2022.

[10] Tao Li, Vivek Gupta, Maitrey Mehta, and Vivek Srikumar. A logic-driven framework for consistency of neural models. *arXiv preprint arXiv:1909.00126*, 2019.

[11] Akari Asai and Hannaneh Hajishirzi. Logic-guided data augmentation and regularization for consistent question answering. *arXiv preprint arXiv:2004.10157*, 2020.

[12] Eric Mitchell, Joseph J Noh, Siyan Li, William S Armstrong, Ananth Agarwal, Patrick Liu, Chelsea Finn, and Christopher D Manning. Enhancing self-consistency and performance of pretrained language models through natural language inference. *arXiv preprint arXiv:2211.11875*, 2022.

[13] Alex Wang, Kyunghyun Cho, and Mike Lewis. Asking and answering questions to evaluate the factual consistency of summaries. *arXiv preprint arXiv:2004.04228*, 2020.

[14] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*, 2020.

[15] Derek Tam, Anisha Mascarenhas, Shiyue Zhang, Sarah Kwan, Mohit Bansal, and Colin Raffel. Evaluating the factual consistency of large language models through news summarization. *arXiv preprint arXiv:2211.08412*, 2022.

[16] Letitia Parcalabescu and Anette Frank. On measuring faithfulness or self-consistency of natural language explanations. *arXiv preprint arXiv:2311.07466*, 2023.

[17] Vamshi Krishna Bonagiri, Sreeram Vennam, Priyanshul Govil, Ponnurangam Kumaraguru, and Manas Gaur. Sage: Evaluating moral consistency in large language models. *arXiv preprint arXiv:2402.13709*, 2024.

[18] Alexios Arvanitis and Konstantinos Kalliris. Consistency and moral integrity: A self-determination theory perspective. *Journal of Moral Education*, 49(3):316–329, 2020.

[19] Faisal Hamman, Pasan Dissanayake, Saumitra Mishra, Freddy Lecue, and Sanghamitra Dutta. Quantifying prediction consistency under fine-tuning multiplicity in tabular LLMs. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=AXJnqocQpm.

[20] Juan Felipe Gomez, Caio Machado, Lucas Monteiro Paes, and Flavio Calmon. Algorithmic arbitrariness in content moderation. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 2234–2253, 2024.

[21] Harsh Raj, Domenic Rosati, and Subhabrata Majumdar. Measuring reliability of large language models through semantic consistency. *arXiv preprint arXiv:2211.05853*, 2022.

[22] Ella Rabinovich, Samuel Ackerman, Orna Raz, Eitan Farchi, and Ateret Anaby-Tavor. Predicting question-answering performance of large language models through semantic consistency. *arXiv preprint arXiv:2311.01152*, 2023.

[23] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*, 2023.

[24] Yukun Zhao, Lingyong Yan, Weiwei Sun, Guoliang Xing, Shuaiqiang Wang, Chong Meng, Zhicong Cheng, Zhaochun Ren, and Dawei Yin. Improving the robustness of large language models via consistency alignment. *arXiv preprint arXiv:2403.14221*, 2024.

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

[26] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781, 2020.

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

[28] Jennifer Hsia, Afreen Shaikh, Zora Zhiruo Wang, and Graham Neubig. Ragged: Towards informed design of scalable and stable rag systems. In *Forty-second International Conference on Machine Learning*.

[29] Kepu Zhang, Zhongxiang Sun, Weijie Yu, Xiaoxue Zang, Kai Zheng, Yang Song, Han Li, and Jun Xu. Qe-rag: A robust retrieval-augmented generation benchmark for query entry errors. *arXiv preprint arXiv:2504.04062*, 2025.

[30] Zhibo Hu, Chen Wang, Yanfeng Shu, Hye-Young Paik, and Liming Zhu. Prompt perturbation in retrieval-augmented generation based large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1119–1130, 2024.

[31] Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. RA-DIT: Retrieval-augmented dual instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=22OTbutug9.

[32] Xinze Li, Sen Mei, Zhenghao Liu, Yukun Yan, Shuo Wang, Shi Yu, Zheni Zeng, Hao Chen, Ge Yu, Zhiyuan Liu, Maosong Sun, and Chenyan Xiong. RAG-DDR: Optimizing retrieval-augmented generation using differentiable data rewards. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Pnktu2PBXD.

[33] Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pages 9112–9141, 2024.

[34] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Rwhi91ideu.

[35] Zilong Ryan Wang, Zifeng Wang, Long Le, Huaixiu Steven Zheng, Swaroop Mishra, Vincent Perot, Yuwei Zhang, Anush Mattapalli, Ankur Taly, Jingbo Shang, et al. Speculative rag: Enhancing retrieval augmented generation through drafting. In *International Conference on Learning Representations*, volume 2025, pages 18483–18505, 2025.

[36] Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. 2024.

[37] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[38] Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. In *International Conference on Learning Representations*, volume 2026, pages 2548–2581, 2026.

[39] Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Xinwei Long, Ermo Hua, Biqing Qi, et al. Ttrl: Test-time reinforcement learning. *Advances in Neural Information Processing Systems*, 38:131459–131483, 2026.

[40] Mengqi Li, Lei Zhao, Anthony Man-Cho So, Ruoyu Sun, and Xiao Li. A model can help itself: Reward-free self-training for llm reasoning. *arXiv preprint arXiv:2510.18814*, 2025.

[41] Ruishuo Chen, Yu Chen, Zhuoran Li, and Longbo Huang. PowerFlow: Unlocking the dual nature of LLMs via principled distribution matching. In Tong Zhang, Miroslav Dudik, Martin Jaggi, Alekh Agarwal, Sharon Li, Dale Schuurmans, Jerry Zhu, Felix Berkenkamp, Hanze Dong, and Alberto Bietti, editors, *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 13700–13725. PMLR, 06–11 Jul 2026. URL https://proceedings.mlr.press/v306/chen26k.html.

[42] John C Gower and Pierre Legendre. Metric and euclidean properties of dissimilarity coefficients. *Journal of classification*, 3(1):5–48, 1986.

[43] Yapei Chang, Yekyung Kim, Michael Krumdick, Amir Zadeh, Chuan Li, Chris Tanner, and Mohit Iyyer. BLEUBERI: BLEU is a surprisingly effective reward for instruction following. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=bEYzeItoOH.

[44] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[45] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

[46] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*, 2020.

[47] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

[48] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. Eli5: Long form question answering. In *Proceedings of ACL 2019*, 2019.

[49] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

[50] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

[51] Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vassilis Plachouras, Tim Rockt"aschel, and Sebastian Riedel. KILT: a Benchmark for Knowledge Intensive Language Tasks. 2020.

[52] Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025.

[53] Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. Inference scaling for long-context retrieval augmented generation. *arXiv preprint arXiv:2410.04343*, 2024.

[54] Yuetong Zhao, Hongyu Cao, Xianyu Zhao, and Zhijian Ou. An empirical study of retrieval augmented generation with chain-of-thought. In *2024 IEEE 14th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 436–440. IEEE, 2024.

[55] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

[56] Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*, 2023.

[57] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[58] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36: 68539–68551, 2023.

[59] Aditi Singh, Abul Ehtesham, Saket Kumar, Tala Talaei Khoei, and Athanasios V Vasilakos. Agentic retrieval-augmented generation: A survey on agentic rag. *arXiv preprint arXiv:2501.09136*, 2025.

[60] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[61] Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

[62] Benoit Courty, Victor Schmidt, Sasha Luccioni, Goyal-Kamal, MarionCoutarel, Boris Feld, Jérémy Lecourt, LiamConnell, Amine Saboni, Inimaz, supatomic, Mathilde Léval, Luis Blanche, Alexis Cruveiller, ouminasara, Franklin Zhao, Aditya Joshi, Alexis Bogroff, Hugues de Lavoreille, Niko Laskaris, Edoardo Abati, Douglas Blank, Ziyao Wang, Armin Catovic, Marc Alencon, Michał Stęchły, Christian Bauer, Lucas Otávio N. de Araújo, JPW, and MinervaBooks. mlco2/codecarbon: v2.4.1, May 2024. URL `https://doi.org/10.5281/zenodo.11171501`.

[63] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):60, 2019.

[64] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. Audio augmentation for speech recognition. In *Proc. Interspeech 2015*, pages 3586–3589, 2015.

[65] Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for nlp. In *Findings of the association for computational linguistics: ACL-IJCNLP 2021*, pages 968–988, 2021.

[66] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=1PL1NIMMrw`.

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

[68] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

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

A Appendix

A.1 Broader Impact 16
A.2 Generating Paraphrased and Semantically Equivalent Queries 16
A.3 Measuring Information Consistency Using LLM-as-a-Judges 17
A.4 BLEU: An n-gram Based Evaluation Metric 19
A.5 Training Configuration 20
A.6 Propagation of Retrieval Inconsistency to End-to-End Consistency 20
A.7 Comparison Between PS-GRPO and Standard GRPO 21
A.8 Con-RAG outperforming SFT for Short-Form QA 21
A.9 Training Cost–Performance Trade-offs under (κ, s) Subsampling 22
A.10 Semantic Similarity Reward: BLEU vs. BERTScore 22
A.11 Why Does Consistency Training Improve Accuracy? 23
A.12 Out-of-Distribution Evaluations 24
A.13 Choosing the KL Coefficient Hyperparameter β 24
A.14 Robustness to Retrieval Depth and Decoding Strategy 25
A.15 Training Cost under Distribution Drift 26

A.1 Broader Impact

This work aims to improve the reliability of retrieval-augmented generation (RAG) systems by reducing inconsistency across semantically equivalent queries. More consistent RAG systems can increase user trust and reduce the risk of confusion or misinformation in high-stakes applications such as healthcare, finance, and legal assistance, where users expect the same guidance regardless of how a question is phrased. This becomes increasingly important as LLMs are deployed in agentic systems that chain reasoning with calls to external tools such as web search: an inconsistent answer at one step can propagate through subsequent actions, and retrieval is often provided by third-party services that cannot be modified, so robustness must come from the model itself.

At the same time, consistency is not correctness. A system that is consistently wrong may appear more authoritative than one that is inconsistent, and improved consistency may increase users' reliance on model outputs. Consistency optimization should therefore be paired with continued evaluation of accuracy, as we do throughout this work. In open-ended settings, enforcing agreement across paraphrases may also reduce legitimate diversity in responses, such as alternative valid perspectives, which may not always be desirable. We do not anticipate novel ethical risks beyond those commonly associated with large language models, but encourage future work on jointly addressing consistency, accuracy, and robustness in real-world deployments.

A.2 Generating Paraphrased and Semantically Equivalent Queries

For each query q_0, we use LLaMA-3.1-70B to generate n paraphrases $\mathcal{P}(q_0) = \{p_1, \ldots, p_n\}$. To ensure answerability, we provide the ground truth answer as part of the prompt and instruct the model to generate paraphrases that preserve the exact meaning such that each paraphrase can be answered in the same way. In our early experiments, we found paraphrases generated without the gold answer degraded the quality of paraphrases generated: they remained similar to the original query but slightly changed the implied answer or introduced new answerable variants. This directly harmed the validity of our consistency evaluation. Including the gold answer anchored the paraphrase generation to the same underlying fact. For example, given the original question "*What nationality was Person X?*" (answer: *American*), a paraphrase generated without the gold answer was "*What was the ethnic background of Person X?*" Providing the gold answer resolved such issues, and in practice we observed no cases where the gold answer leaked into the paraphrases.

Table 6: **Effect of gold answers on paraphrase quality** (HotpotQA). Accuracy on original queries and paraphrased queries generated with and without access to the gold answer. Including the gold answer during paraphrase generation preserves semantic equivalence, while omitting it leads to substantial accuracy degradation.

Setting	Query Type	EM	F1	RM
–	Original Queries	37.0	44.1	42.0
With Gold Answer	Paraphrased Queries	36.4	43.5	42.4
Without Gold Answer	Paraphrased Queries	27.4	36.1	35.4

To further demonstrate this, we generate paraphrases on HotpotQA both with and without ground truth and compare accuracy across original queries and paraphrased queries. As shown in Table 6, with the gold answer included, paraphrased queries maintain nearly identical accuracy to the original queries (e.g., EM drops from 37.0 → 36.4). In contrast, when the gold answer is omitted, accuracy on paraphrased queries drops sharply (EM: 37.0 → 27.4), indicating that many paraphrases drift semantically. Importantly, this does not affect the model being evaluated. Paraphrases are generated offline, and the ground truth is never shown to the RAG system.

We note that for long-form QA tasks (e.g., ELI5, NQ) we do not include ground truth during paraphrase generation, and this issue does not arise. See prompt used for short-form and long form QA tasks below:

`Paraphrasing - Short-form QA`

You are given an input sentence. Your task is to generate `n` diverse paraphrases of this sentence. You can paraphrase by using synonyms, changing sentence structure, or rephrasing in any other way, but each paraphrase should preserve the original meaning. Each paraphrase you create must be answerable by the exact same answer provided below.
Format your output as follows:
`<paraphrase1>` paraphrased sentence 1 `</paraphrase1>`
`<paraphrase2>` paraphrased sentence 2 `</paraphrase2>`
...
`<paraphrasen>` paraphrased sentence `n` `</paraphrasen>`
Input sentence: `{sentence}` **Required answer:** `{answer}`
Please return only the paraphrases in the specified format.

`Paraphrasing - Long-form QA`

You are given an input question sentence. Your task is to generate `n` diverse paraphrases of this question. You can paraphrase by using synonyms, changing sentence structure, or rephrasing in any other way, but each paraphrase should preserve the original question meaning and lead to similar answers.
Format your output as follows:
`<paraphrase1>` paraphrased sentence 1 `</paraphrase1>`
`<paraphrase2>` paraphrased sentence 2 `</paraphrase2>`
...
`<paraphrasen>` paraphrased sentence `n` `</paraphrasen>`
Input question: `{sentence}`
Please return only the paraphrases in the specified format.

A.3 Measuring Information Consistency Using LLM-as-a-Judges

To measure information semantic consistency between generated outputs across paraphrased queries, we employ LLM-as-a-judge as part of our evaluation pipeline. This judge is designed to determine whether two answers convey the same core information, even if they differ in surface form. For all LLaMA-3.1-8B experiments, we use LLaMA-3.3-70B as the evaluator. For all Qwen-2.5-3B experiments, we use GPT-4o as the evaluator. An example evaluation prompt is shown below:

Table 7: **Accuracy across datasets and query variants** (LLaMA-3.1-8B). We report accuracy for original queries, synthetically generated paraphrased queries, and paraphrased queries with fixed retrieval. Across all settings, accuracy remains relatively similar, indicating that paraphrasing and retrieval shifts have limited effect on final answer correctness on average. See result for Qwen-2.5-3B model in Table 9.

Short-form & Multi-hop QA: Accuracy (%)									
Dataset	**Original Queries**			**Paraphrased Queries**			**Paraphrased (Fixed Docs)**		
	EM	F1	RM	EM	F1	RM	EM	F1	RM
TriviaQA	56.0	66.1	74.0	55.0	64.4	73.3	58.7	67.3	75.0
HotpotQA	37.0	44.1	42.0	36.4	43.5	42.4	33.7	40.7	39.4
2Wiki	28.0	33.9	37.0	25.9	31.3	32.7	26.9	31.7	33.3
MuSiQue	8.0	15.3	12.0	8.3	14.1	11.0	11.0	17.5	15.0

Long-form QA: Accuracy (%)						
Dataset	**Original Queries**		**Paraphrased Queries**		**Paraphrased (Fixed Docs)**	
	ROUGE	LLM-Acc	ROUGE	LLM-Acc	ROUGE	LLM-Acc
ELI5	21.9	74.0	20.7	71.3	20.8	70.3
NQ	26.9	61.0	25.2	58.8	25.2	57.9

Table 8: **Disentangling sources of inconsistency in RAG systems** (Qwen-2.5-3B). Retriever consistency is low across datasets, suggesting that paraphrased queries often retrieve non-overlapping documents. This introduces context variability that is reflected in the end-to-end consistency scores. Fixing retrieval improves consistency, but variation remains, revealing the generator's sensitivity to input phrasing even with identical evidence.

Dataset	**End-to-End Consistency**		**Generator (LLM) Consistency**		**Retriever Consistency**
	Lexical	Inform.	Lexical	Inform.	Jaccard Overlap
TriviaQA	47.9	73.0	58.6	87.5	32.5
HotpotQA	32.7	63.6	48.0	77.3	46.0
2Wiki	32.3	62.6	44.6	70.7	52.4
MuSiQue	25.7	49.5	45.7	67.3	36.6
ElI5	6.6	35.3	14.4	62.3	27.1

`Consistency Evaluation - Long-form QA`

You are an evaluator assessing two different answers that were generated for the same question. Your task is to determine if the two answers are consistent with each other.
Consider them consistent if they present the same core information about the underlying question. Consider them inconsistent if they provide different information, present different facts, or address the underlying question in different ways.
Based on this, reply with only `yes` or `no`. Do not output anything else.
Answer 1: `{output_i}`
Answer 2: `{output_j}`
Are these two answers consistent? (yes/no). Response:

`Consistency Evaluation - Short-form QA`

Given the following two outputs sentences, reply with `yes` if the two sentences are consistent with each other, or `no` if they are not. Do not output anything else.
Sentence 1: `{output_i}`
Sentence 2: `{output_j}`
Are these sentences consistent? (yes/no). Response:

Table 9: **Accuracy across datasets and query variants** (Qwen-2.5-3B). We report accuracy for original queries, synthetically generated paraphrased queries, and paraphrased queries with fixed retrieval. Across all settings, accuracy remains relatively similar, indicating that paraphrasing and retrieval shifts have limited effect on final answer correctness on average.

Short-form & Multi-hop QA: Accuracy (%)									
Dataset	**Original Queries**			**Paraphrased Queries**			**Paraphrased (Fixed Docs)**		
	EM	F1	RM	EM	F1	RM	EM	F1	RM
TriviaQA	42.0	50.7	58.0	46.3	54.1	64.3	43.0	51.1	62.3
HotpotQA	20.0	28.3	37.0	20.9	27.7	38.4	18.2	26.4	38.4
2Wiki	13.0	20.4	36.0	11.1	19.8	34.7	12.5	20.2	32.3
MuSiQue	4.0	10.0	9.0	6.0	9.9	8.0	5.3	9.7	7.7

Long-form QA: Accuracy (%)						
Dataset	**Original Queries**		**Paraphrased Queries**		**Paraphrased (Fixed Docs)**	
	ROUGE	LLM-Acc	ROUGE	LLM-Acc	ROUGE	LLM-Acc
ElI5	22.1	38.0	21.5	37.3	20.8	35.7

Table 10: **Comparison between Con-RAG and Baselines** (Short-form QA, Qwen-2.5-3B). Lexical consistency measured via BLEU score while information consistency measured using an LLM-as-a-judge. Con-RAG is trained with a group-similarity reward plus an accuracy reward (no KL), and consistently yields higher end-to-end and generator-only consistency while also improving accuracy over original queries.

		Accuracy (%)			**End-to-End Consistency (%)**		**Generator (LLM) Consistency (%)**	
Dataset	**Method**	EM	F1	RM	Lexical	Inform.	Lexical	Inform.
TriviaQA	RAG	42.0	50.7	58.0	47.9	73.0	58.6	87.5
	DRAG	42.0	50.7	58.0	47.9	73.5	58.6	84.7
	CoT-RAG	37.0	44.5	61.0	41.1	72.3	52.2	82.3
	SFT	35.0	40.4	43.0	53.3	72.2	73.4	85.0
	Con-RAG	**60.0**	**66.0**	**68.0**	**67.1**	**81.8**	**80.5**	**89.5**
HotpotQA	RAG	20.0	28.3	37.0	32.7	63.6	48.0	77.3
	DRAG	20.0	28.3	37.0	32.7	64.3	48.0	76.8
	CoT-RAG	29.0	32.8	37.0	28.5	63.6	35.7	71.2
	SFT	30.0	35.4	32.0	63.3	77.1	74.5	85.7
	Con-RAG	**36.0**	**43.1**	**38.0**	**64.6**	**78.2**	**77.8**	**86.7**
MuSiQue	RAG	4.0	10.0	9.0	25.7	49.5	45.7	67.3
	DRAG	4.0	10.0	9.0	25.7	50.3	45.7	69.2
	CoT-RAG	5.0	10.9	9.0	18.1	52.0	26.5	62.0
	SFT	25.0	30.6	27.0	57.7	65.3	69.8	77.2
	Con-RAG	**27.0**	**31.9**	**28.1**	**69.8**	**70.1**	**70.4**	**82.0**
2Wiki	RAG	13.0	20.4	36.0	32.3	62.6	44.6	70.7
	DRAG	13.0	20.4	36.0	32.3	63.0	44.6	70.9
	CoT-RAG	23.0	27.0	30.0	23.5	62.3	32.7	67.0
	SFT	**37.0**	**38.9**	**38.0**	**70.9**	75.8	**84.8**	86.9
	Con-RAG	**37.0**	38.4	37.0	68.2	**76.6**	**84.8**	**89.1**

A.4 BLEU: An n-gram Based Evaluation Metric

The BLEU (Bilingual Evaluation Understudy [60]) score is a standard metric for assessing the quality of machine translation. It quantifies the degree of overlap between a system-generated translation and one or more human reference translations. The score relies on modified n-gram precision ($n \in \{1, 2, 3, 4\}$), together with a brevity penalty (BP) that discourages excessively short outputs:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \quad \text{BP} = \begin{cases} 1 & \text{if } c > r, \\ \exp\left(1 - \frac{r}{c}\right) & \text{if } c \leq r, \end{cases}$$

Table 11: **Comparison between Con-RAG and baselines** (Long-form QA). Con-RAG is trained using only the group-similarity reward with a small KL regularizer (no accuracy supervision). Despite no ground-truth, it achieves the best end-to-end and generator consistency and also maintains answer quality over baselines, whereas SFT on reference answers underperforms in this open-ended setting (Qwen-2.5-3B).

		Accuracy (%)		**End-to-End Consistency (%)**		**Generator (LLM) Consistency (%)**	
Dataset	**Method**	ROUGE	LLM-Acc	Lexical	Inform.	Lexical	Inform.
ELI5	RAG	22.1	38.0	6.6	35.3	14.4	62.3
	DRAG	22.1	38.0	6.6	35.3	14.4	63.8
	CoT-RAG	21.1	36.0	4.9	34.0	9.6	55.5
	SFT	**24.3**	36.0	5.4	17.2	7.0	19.0
	CoG	23.6	40.2	6.2	20.4	10.3	27.3
	CAT	23.8	43.6	5.8	25.5	10.1	29.3
	Con-RAG	22.6	**58.0**	**9.3**	**42.8**	**17.9**	**67.5**

where p_n denotes the modified n-gram precision, w_n are the associated weights, c is the length of the candidate translation, and r is the length of the closest reference. When multiple references are given, BLEU counts n-gram matches against all references and uses the maximum match count for each n-gram. Each n-gram level in BLEU captures progressively deeper aspects of linguistic quality. Unigrams (n=1) assess word choice or adequacy, indicating whether the candidate includes the correct content words. Bigrams (n=2) begin to reflect local fluency by capturing short-range word ordering. Trigrams (n=3) provide signals about phrase-level coherence, identifying whether multi-word chunks align with natural phrasing. 4-grams (n=4) enforce sentence-level fluency by requiring longer, contiguous sequences to match the reference.

A.5 Training Configuration

All Con-RAG models are trained with full-parameter fine-tuning in TRL on 8×NVIDIA A100 GPUs, using DeepSpeed ZeRO-3 and bf16 mixed precision. We use AdamW with a learning rate of $(10^{-5}, 10^{-6})$ and a linear schedule without warmup, train for 4 epochs, and set the GRPO clipping parameter to ϵ=0.2. Rewards use $(\kappa = 3, s = 1)$. Short-form and multi-hop QA use BLEU-1 plus a token-F1 accuracy reward (α=γ=1, β=0); long-form QA uses BLEU-2 with consistency only (γ=0, β=0.05). Training rollouts are sampled at temperature 1.0 and top-p 1.0, with a maximum completion length of 16 tokens for short-form and 64 tokens for long-form QA; evaluation uses greedy decoding with n=5 paraphrases per query. Full configurations and scripts are available at `https://github.com/CapitalOne-Research/con-rag`.

A.6 Propagation of Retrieval Inconsistency to End-to-End Consistency

Retrieval variation propagates into output inconsistency, and Con-RAG's gains concentrate where it is most prevalent. The fixed-retrieval experiment in Table 1 shows that generator sensitivity remains when retrieval is controlled, but does not by itself explain how much retriever inconsistency propagates into final output inconsistency. To quantify this relationship, we conduct a per-query analysis on NQ. For each canonical query, we compute (i) retriever Jaccard overlap across its paraphrase set, (ii) generator consistency under fixed retrieval, and (iii) end-to-end consistency, then bin queries by retrieval overlap and report conditional consistency (see Table 12). Let Δ be the difference between generator consistency, where every paraphrase is conditioned on identical evidence, and end-to-end consistency, where each paraphrase retrieves its own documents. Δ measures the share of output inconsistency that disappears once evidence is equalized.

For standard RAG, end-to-end information consistency rises from 41.5% to 64.9% as retrieval overlap increases, while fixed-retrieval generator consistency remains nearly unchanged ($67.0 - 69.2\%$). Accordingly, the retrieval-associated gap decreases from 25.5 points under low overlap to only 4.3 points under high overlap. This directly shows that retrieval variation propagates into output inconsistency when paraphrases retrieve different evidence.

Con-RAG provides its largest improvement in the low-overlap setting, increasing end-to-end consistency from 41.5% to 55.2% and reducing the gap from 25.5 to 19.6 points. Low and medium-overlap

Table 12: **Retrieval-induced inconsistency propagation** (NQ, LLaMA-3.1-8B). Queries are binned by retriever consistency (Jaccard overlap across the paraphrase set). We report LLM-Judge consistency, and the propagation gap Δ = Gen. − E2E, i.e., the share of output inconsistency that disappears once evidence is equalized across paraphrases. For RAG, Δ shrinks from 25.5 to 4.3 points as retrieval overlap increases, showing that retrieval variation propagates into output inconsistency. Con-RAG's largest end-to-end gains occur in the low-overlap regime, where retrieval variation is the dominant source of inconsistency.

Retriever Jaccard	% Queries	Method	E2E Cons.	Gen. Cons.	Δ (Gen.−E2E)
0.0–0.2 (low)	31%	RAG	41.5	67.0	25.5
		Con-RAG	**55.2**	**74.8**	**19.6**
0.2–0.5 (mid)	45%	RAG	54.0	68.4	14.4
		Con-RAG	**59.8**	**75.6**	15.8
0.5–1.0 (high)	24%	RAG	64.9	69.2	4.3
		Con-RAG	**67.5**	**76.1**	8.6

queries constitute 76% of the evaluation set, indicating that Con-RAG is most beneficial precisely where retrieval variation is prevalent. In the high-overlap bin, where retrieval variation is minimal, RAG and Con-RAG perform comparably on end-to-end consistency (64.9 vs. 67.5), as expected, there is little retrieval-induced variability left to absorb, and Con-RAG's advantage there appears at the generator level (69.2 $\rightarrow$ 76.1). Con-RAG also improves fixed-retrieval generator consistency in every bin.

A.7 Comparison Between PS-GRPO and Standard GRPO

PS-GRPO does not require any ground-truth references to compute rewards. This is a central property of PS-GRPO: the model improves information consistency purely from group, without using reference ground truth answers. This makes the method applicable to long-form and open-ended QA settings where single gold answers are unavailable or insufficient. We compare with standard GRPO, with BLEU between the generated output and ground-truth answers as reward. Empirically, PS-GRPO outperforms GRPO with ground-truth on both consistency and accuracy (see results in Table 13). The learning signal from mutual agreement across paraphrases seem to be more effective than simply pushing each response toward a single reference.

Table 13: **Comparison between PS-GRPO and standard GRPO on Natural Questions (NQ).** Standard GRPO uses BLEU reward between generated output and ground-truth answers, while PS-GRPO (Con-RAG) uses only group similarity rewards without ground-truth supervision. PS-GRPO outperforms GRPO with ground-truth on both consistency and accuracy.

Method	Accuracy (%)		End-to-End Cons. (%)		Generator Cons. (%)	
	ROUGE	LLM-Acc	Lexical	Inform.	Lexical	Inform.
GRPO (BLEU reward – ground truth supervision)	37.4	52.0	19.2	47.5	29.6	60.5
PS-GRPO (Con-RAG – no ground truth supervision)	23.5	**59.0**	**22.8**	**59.3**	**32.3**	**75.5**

A.8 Con-RAG outperforming SFT for Short-Form QA

SFT treats each paraphrase-answer pair independently and minimizes token-level cross-entropy with a single reference answer (or ground truth). This encourages the model to reproduce the surface form of that reference and penalizes any deviation (even when multiple phrasings are equally correct). For instance, a date like "November 16th, 2025" may be expressed accurately in several valid forms ("*Nov. 16, 2025*," "*16 November 2025,*" etc.), yet SFT penalizes all but the exact reference. This also explains why the F1-based accuracy reward was most effective in our experiments compared to Exact match (EM) and Relaxed Match (RM) as it provides a more flexible alignment signal that tolerates variations in phrasing (see Table 5). In contrast, Con-RAG explicitly optimizes a group-level objective over paraphrase sets. Outputs are rewarded not only for matching the ground truth but also for being consistent with outputs generated for other paraphrases of the same underlying question. This means

Table 14: **Natural Questions (NQ) results across (κ, s) settings.** We report accuracy, consistency, and computational cost while varying the number of sampled paraphrases κ and rollouts per paraphrase s.

Dataset	Params		Accuracy		E2E Consistency		Gen. Consistency		Compute	
	κ	s	ROUGE	LLM-Acc	Lex.	Inform.	Lex.	Inform.	CPU (W)	GPU (W)
NQ	1	1	25.2	60.0	21.3	60.5	30.7	70.8	118	441
	2	1	24.1	59.0	31.3	59.3	42.7	74.0	120	457
	5	1	24.3	60.0	28.4	59.3	38.5	73.0	123	462
	5	5	25.6	63.0	23.5	59.8	33.9	76.2	135	492

that alternative answer formulations can still receive positive reward if they remain consistent across paraphrases, rather than being penalized for surface-level differences.

One could have a variant of our approach for SFT which has a loss term that penalizes inconsistency across paraphrases. However, recent evidence further supports our design choice of using an RL approach rather than SFT to optimize consistency: SFT Memorizes, RL Generalizes [61] shows that SFT largely encourages memorization of surface patterns, causing models to overfit to the exact phrasing and structure of training examples, while RL, with outcome-level rewards, learns transferable rules that robustly generalize to unseen variants. Thus, Con-RAG's gains over SFT align with the broader finding that RL is inherently better suited for learning structures that must remain stable across input variations.

A.9 Training Cost–Performance Trade-offs under (κ, s) Subsampling

To assess the computational trade-offs introduced by the relaxed group-similarity reward, we performed a controlled sweep over the subsampling parameters κ (number of paraphrases sampled) and s (rollouts per paraphrase) on the NQ (Natural Questions) dataset while logging CPU/GPU energy and power using the CodeCarbon python package [62]. Results are summarized in Table 14. We find that Con-RAG's performance is stable across a wide range of (κ, s) settings: increasing κ and s yields only mild fluctuations in accuracy and consistency metrics. However, the computational footprint grows noticeably with larger subsampling budgets. These results highlight that the reward signal is not highly sensitive to these hyperparameters, meaning practitioners can safely adopt low-cost settings which already provide strong consistency gains at a fraction of the compute. For high-resource settings, larger (κ, s) provide marginal additional improvements but at a noticeably higher cost, defining a compute-quality frontier for deployment scenarios with fixed budgets.

A.10 Semantic Similarity Reward: BLEU vs. BERTScore

Our default group similarity reward is instantiated with BLEU, a lexical metric. A natural question is whether a semantic similarity function yields better information consistency, and whether the gains of Con-RAG are driven by surface-level lexical convergence (i.e., the policy concentrating onto canonical, high-overlap responses) rather than agreement in information content. To study this, we replace BLEU with BERTScore as the similarity function $\mathrm{sim}(\cdot, \cdot)$ in the group similarity reward on NQ (LLaMA-3.1-8B), keeping all other training settings identical to our default long-form configuration (κ=3, s=1). As in Appendix A.9, we log CPU and GPU power using the CodeCarbon python package [62]. Results are summarized in Table 15.

Information consistency is not a by-product of lexical convergence. BERTScore achieves comparable information consistency but with lower lexical similarity scores compared to BLEU, which is expected, since BERTScore permits semantically equivalent but lexically diverse outputs. With BERTScore, lexical agreement drops markedly (E2E $22.8 \rightarrow 18.5$; Gen. $32.3 \rightarrow 27.4$), while LLM-Judge consistency and accuracy are essentially unchanged (E2E $59.3 \rightarrow 60.1$; Gen. $75.5 \rightarrow 74.8$; LLM-Acc $59.0 \rightarrow 60.0$). If the gains of Con-RAG were driven by surface-level mode concentration, they should depend on a reward that rewards surface overlap. Instead, when the reward tolerates lexical diversity, the model retains phrasing diversity while preserving content agreement. This is consistent with our observation that Con-RAG's LLM-Judge consistency substantially exceeds its lexical consistency across datasets (Tables 2 & 3).

Table 15: **Effect of a semantic similarity reward on Con-RAG: BLEU vs. BERTScore** (NQ, LLaMA-3.1-8B). We replace BLEU with BERTScore as the similarity function in the group similarity reward. BERTScore achieves comparable LLM-Judge consistency and accuracy, but with lower lexical consistency, since it permits semantically equivalent but lexically diverse outputs. This comes at a substantially higher compute cost, as each comparison requires a BERT encoder forward pass. CPU and GPU power are logged using CodeCarbon [62].

Reward Metric	Accuracy (%)		End-to-End Cons. (%)		Generator Cons. (%)		Compute	
	ROUGE	LLM-Acc	Lexical	Inform.	Lexical	Inform.	CPU (W)	GPU (W)
BLEU (default)	23.5	59.0	**22.8**	59.3	**32.3**	**75.5**	121	459
BERTScore	**23.8**	**60.0**	18.5	**60.1**	27.4	74.8	141	629

Compute trade-off. The comparable performance comes at a significantly higher computational cost: GPU power increases from 459W to 629W (37%) and CPU power from 121W to 141W, since each comparison requires a BERT encoder forward pass. In contrast, BLEU is purely string-based and each comparison is negligible in cost. This matters in our setting, where group similarity rewards must be computed hundreds of times per optimization step. We view this as a cost-effectiveness spectrum: BLEU offers the best efficiency-to-performance ratio; BERTScore trades compute for tolerance of lexical diversity; and an LLM-as-a-judge would enable richer, prompt-specified notions of consistency, but at prohibitive training-time cost. Our framework is agnostic to the exact similarity function, allowing practitioners to substitute stronger semantic rewards (e.g., BERTScore, an LLM critic) depending on their compute budget.

A.11 Why Does Consistency Training Improve Accuracy?

Our goal is to improve information consistency without compromising accuracy; accuracy gains, particularly those obtained without ground-truth supervision, are a welcome byproduct rather than the primary objective. These gains are not simply a consequence of reduced output diversity. We hypothesize they arise from two complementary mechanisms.

(1) Implicit data augmentation. By tying multiple paraphrased inputs of the same query together during training, our approach acts as a form of regularization: the objective pushes the model to capture the invariant semantic structure across paraphrases rather than their lexical differences. This is analogous to data augmentation, where label-preserving transformations such as cropping and flipping in vision [63], noise injection in speech [64], and paraphrasing in NLP [65] improve generalization by exposing the model to diverse realizations of the same underlying input. As a result, the model becomes more robust to paraphrasing.

(2) Consistency as a regularizer against hallucination. Correct answers are inherently more reproducible across paraphrases than hallucinations: a hallucinated answer is unlikely to appear consistently across different phrasings and retrieved documents. This premise underlies self-consistency decoding [66], semantic-entropy hallucination detection [23], and majority-vote rewards [39]. Rewarding cross-paraphrase agreement therefore favors content grounded in evidence shared across retrievals and filters out noise, similar to a "wisdom of crowds" effect. The KL term prevents the model from collapsing to trivial but consistent outputs.

Empirically, consistency-only training (α=1, γ=0) already improves accuracy over RAG on TriviaQA (EM 51.5 vs. 42.0; Table 5), and on ELI5, Con-RAG trained without any ground truth achieves the highest LLM-judged accuracy of all methods (78.0 vs. 74.0 for RAG; Table 3). On NQ, accuracy is preserved within two points of RAG (59.0 vs. 61.0) alongside large consistency gains.

Consistent but wrong outputs. Our method can lead to consistent but wrong outputs on certain inputs. Even in such cases, consistency is preferable to being both inconsistent and wrong, since the latter produces more unpredictable and contradictory outputs. In our experiments, we do not observe this trade-off dominating in practice: consistency improvements are accompanied by improved or preserved accuracy relative to RAG. Future work could more deeply characterize when consistency and accuracy align or diverge, especially in long-form, open-ended settings.

Table 16: **Out-of-distribution generalization of Con-RAG on long-form QA** (LLaMA-3.1-8B). Con-RAG is trained on the dataset in the *Train* column and evaluated on the dataset in the *Eval* column (ID: in-distribution; OOD: out-of-distribution). Transferred models preserve accuracy close to the RAG baseline on the unseen dataset while still improving both end-to-end and generator consistency, indicating a transferable consistency behavior rather than dataset-specific output concentration.

			Accuracy (%)		**End-to-End Consistency (%)**		**Generator (LLM) Consistency (%)**	
Eval	**Method**	**Train**	ROUGE	LLM-Acc	Lexical	Inform.	Lexical	Inform.
ELI5	RAG	–	21.9	74.0	8.6	62.8	15.1	74.2
	Con-RAG	ELI5 (ID)	**24.2**	**78.0**	**14.6**	**72.7**	**21.7**	**80.8**
	Con-RAG	NQ (OOD)	23.1	74.0	12.8	69.5	19.4	77.6
NQ	RAG	–	**26.9**	**61.0**	14.2	53.2	22.4	68.2
	Con-RAG	NQ (ID)	23.5	59.0	**22.8**	**59.3**	**32.3**	**75.5**
	Con-RAG	ELI5 (OOD)	22.8	57.0	19.5	56.8	28.9	72.1

A.12 Out-of-Distribution Evaluations

A natural concern with a consistency-based objective is that the policy may improve group similarity rewards by locking onto a set of generic, dataset-specific safe outputs rather than learning a transferable consistency behavior. To test this, we conduct out-of-distribution (OOD) evaluations on long-form QA: we train Con-RAG on one long-form dataset and evaluate it on the other (ELI5→NQ and NQ→ELI5). If Con-RAG had collapsed onto dataset-specific outputs, these outputs would carry over to the new distribution, where they would no longer be correct, and accuracy would fall sharply while consistency stayed artificially high.

We do not observe this behavior (see Table 16). On ELI5, the model trained on NQ matches the RAG baseline in LLM-judged accuracy while improving information consistency at both the end-to-end ($62.8 \rightarrow 69.5$) and generator ($74.2 \rightarrow 77.6$) levels. On NQ, the model trained on ELI5 remains close to the RAG baseline in accuracy (57.0 vs. 61.0) while improving information consistency from $53.2 \rightarrow 56.8$ (end-to-end) and $68.2 \rightarrow 72.1$ (generator). Lexical consistency improves over RAG in both directions as well. As expected, in-distribution training yields the largest gains, but the transferred models still improve over the baseline on every consistency metric. This indicates that Con-RAG learns a transferable consistency behavior rather than concentrating on dataset-specific outputs. Training on a mixed-task paraphrase corpus to obtain a single broadly consistent model is a natural extension, and would further reduce the need for per-dataset training.

A.13 Choosing the KL Coefficient Hyperparameter β

A consistency-only objective can in principle be maximized by converging to answers that agree across paraphrases but are trivial or wrong; a KL penalty to the reference policy mitigates this risk. When rewards are verifiable against ground truth, recent GRPO work has progressively weakened this penalty, from β=0.04 [37] to β=0.001 [67], and in several cases removed it entirely [68, 52, 69], since a rule-based reward offers no learned model for the policy to exploit. Our label-free setting lacks this safeguard. Below, we summarize the evidence from our experiments and give practical guidelines for setting β.

When an accuracy term is present, β=0 suffices. In short-form and multi-hop QA, the token-F1 reward anchors outputs to correct content, and we train without KL, following [52]. Even without the accuracy term, consistency-only training on TriviaQA does not collapse: it improves EM over RAG (51.5 vs. 42.0; Table 5). Short answers leave little room for generic outputs, and the paraphrase-level reward still favors the answer that is reproducible across retrieved contexts.

Without ground truth, a small β prevents degenerate consensus. For long-form QA we use β=0.05. Our metrics detect collapse when it occurs: its signature is rising lexical consistency with falling information consistency and accuracy. SFT shows exactly this pattern on NQ (lexical 37.1, LLM-Judge 36.2, LLM-Acc 22.0; Table 3). Con-RAG with β=0.05 shows the opposite pattern, with LLM-Judge consistency well above lexical consistency (59.3 vs. 22.8) and accuracy preserved (59.0 vs. 61.0 for RAG); on ELI5 it achieves the highest LLM-Acc of any method (78.0). The setting is also

Table 17: **Sensitivity of Con-RAG to the number of retrieved documents (top-k)** (NQ, LLaMA-3.1-8B). Accuracy slightly improves with larger k, while consistency mildly decreases as more tangential content is introduced. Con-RAG maintains large consistency gains over the RAG baseline across all k values.

Setting	Accuracy (%)		End-to-End Consistency (%)		Generator (LLM) Consistency (%)	
	ROUGE	LLM-Acc	Lexical	Inform.	Lexical	Inform.
RAG (k=5)	**26.9**	**61.0**	14.2	53.2	22.4	68.2
Con-RAG (k=5, default)	23.5	59.0	**22.8**	**59.3**	**32.3**	**75.5**
Con-RAG (k=10)	24.1	60.0	21.3	57.8	31.0	74.2
Con-RAG (k=15)	24.4	**61.0**	19.7	56.1	29.5	72.8

robust to other design choices: across the (κ, s) sweep, LLM-Acc stays within 59.0–63.0 (Table 14), and replacing BLEU with BERTScore leaves accuracy unchanged (Appendix A.10).

Guidelines. (i) If gold answers are available, include the accuracy term and set β=0. (ii) Without gold answers, start from a small β (we use 0.05) and select it on a small held-out set scored by an LLM judge for accuracy, not by training reward, since the consistency reward increases monotonically as β decreases. (iii) During training, monitor collapse signatures: lexical consistency rising while LLM-Judge consistency or judged accuracy falls, shrinking response length or distinct-n, or rapidly growing KL to the reference. If any appears, increase β.

A.14 Robustness to Retrieval Depth and Decoding Strategy

Sensitivity to the number of retrieved documents (top-k). Con-RAG is designed to be retriever-agnostic. Our RAG pipeline (e5-base-v2 embeddings, top-5 retrieval, 512-token chunks, KILT Wikipedia corpus) was configured to first maximize accuracy over original queries before conducting any consistency analysis on paraphrased queries, mirroring how a RAG system would be set up in practice. This setup is also consistent with related work on RAG evaluation and optimization [53]. To evaluate the sensitivity of Con-RAG to this pipeline parameter, we vary the number of retrieved documents $k \in \{5, 10, 15\}$ on NQ (LLaMA-3.1-8B). Results are summarized in Table 17. Accuracy slightly improves with larger k (more relevant evidence), while consistency shows a mild decrease, as additional noisy and tangential content increases the divergence surface across paraphrased queries. Importantly, Con-RAG maintains large consistency gains over the RAG baseline across all k values (e.g., E2E LLM-Judge consistency of 56.1–59.3 vs. 53.2 for RAG), confirming robustness to the pipeline configuration. Since our method is agnostic to the retriever, we do not expect results to change substantially with different retrieval mechanisms.

Effect of decoding strategy on Con-RAG. In our main experiments, we use deterministic decoding at inference specifically to isolate the inconsistency due to paraphrasing from sampling stochasticity. This is a deliberate methodological choice and is standard practice in consistency evaluation [2, 3]. We note that during training, we use non-deterministic decoding to generate diverse rollouts; deterministic decoding is only used at inference. To evaluate practicality under more common decoding strategies, we compare greedy decoding against top-p sampling (p=0.9), top-k sampling (k=50), and temperature sampling (T=0.5) for Con-RAG on NQ (LLaMA-3.1-8B) (see Table 18). Stochastic decoding introduces a moderate drop in lexical consistency, as expected, since sampling adds variability on top of paraphrase-induced variation. However, LLM-Judge consistency and accuracy remain stable, and Con-RAG remains above the greedy RAG baseline (E2E LLM-Judge 53.2, Gen. LLM-Judge 68.2) under every decoding strategy. Temperature sampling at T=0.5 is closest to greedy decoding and even slightly improves LLM-Acc (60.0 vs. 59.0).

We further evaluate how inference-time stochasticity affects standard RAG by sweeping temperature values $T \in \{0.0, 0.5, 1.0, 2.0\}$ on ELI5 (Qwen-2.5-3B) (see Table 19). Moderate temperature (T=0.5) improves LLM-Judge and lexical consistency compared to deterministic decoding while preserving accuracy, whereas higher temperatures ($T \geq 1.0$) degrade both consistency and accuracy, with outputs at T=2.0 nearly collapsing.

Table 18: **Effect of decoding strategy on Con-RAG** (NQ, LLaMA-3.1-8B). Stochastic decoding moderately reduces lexical consistency, while LLM-Judge consistency and accuracy remain stable. Con-RAG remains above the greedy RAG baseline (E2E LLM-Judge 53.2, Gen. LLM-Judge 68.2) under all decoding strategies.

Decoding	**Accuracy (%)**		**End-to-End Cons. (%)**		**Generator Cons. (%)**	
	ROUGE	LLM-Acc	Lexical	Inform.	Lexical	Inform.
Greedy (T=0.0, default)	**23.5**	59.0	**22.8**	**59.3**	**32.3**	**75.5**
Top-p (p=0.9)	22.8	58.0	18.4	56.8	28.1	72.3
Top-k (k=50)	22.6	57.0	17.9	55.5	27.4	71.8
Temperature (T=0.5)	23.1	**60.0**	20.2	58.1	30.5	74.0

Table 19: **Effect of inference temperature on standard RAG** (ELI5, Qwen-2.5-3B). We vary only the decoding temperature T at inference to study its effect on consistency and accuracy. Moderate temperature (T=0.5) improves LLM-Judge and lexical consistency compared to deterministic decoding (T=0.0), while preserving accuracy. However, higher temperatures ($T \geq 1.0$) degrade both consistency and accuracy, with outputs at T=2.0 nearly collapsing.

T	**Accuracy (%)**		**End-to-End Cons. (%)**		**Generator Cons. (%)**	
	ROUGE	LLM-Acc	Lexical	Inform.	Lexical	Inform.
0.0	**22.1**	38.0	6.6	35.3	14.4	62.3
0.5	21.4	**52.0**	**10.4**	**37.7**	**15.2**	**65.3**
1.0	21.8	48.0	2.5	34.0	5.2	59.5
2.0	6.1	0.0	0.1	2.0	0.2	1.5

A.15 Training Cost under Distribution Drift

A practical concern is the cost of retraining Con-RAG when the query distribution shifts rapidly. We have not studied retraining cadence under drift directly, but several factors reduce the practical cost.

(i) Retraining requires no annotation. Under drift, the dominant recurring cost for supervised alternatives (SFT, CoG, CAT) is producing reference answers for the new distribution. PS-GRPO's group similarity reward is computed from mutual agreement among rollouts across paraphrases, so a refresh needs only new queries and their paraphrases.

(ii) Retraining may not be needed for every shift. Our out-of-distribution results (Table 16) suggest the learned behavior is not distribution-specific: a model trained on NQ and evaluated on ELI5 retains much of its consistency gain over RAG at comparable accuracy. Training on a mixed-task paraphrase corpus to obtain a single, broadly consistent model is a natural extension that would further reduce retraining frequency.

(iii) A refresh can run at the low-cost end of the compute frontier. The subsampling ablation (Table 14) shows the minimal setting is already competitive: (κ=1, s=1) achieves 60.0 LLM-Acc and 70.8 generator LLM-Judge consistency at 441 W GPU power, compared with 63.0 / 76.2 at 492 W for (κ=5, s=5). Distribution drift is a general challenge for every trained component of a RAG pipeline, and retrieval quality and accuracy are likely to degrade before consistency becomes the binding concern. Con-RAG is therefore not uniquely exposed, and its label-free refresh makes it among the cheaper components to keep current.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Abstract and introduction match papers claims. Details in Main Section 2.2

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Discussed in Section 4.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: The paper has no theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Experiment details are in Section 3.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: Code for paraphrase generation, PS-GRPO training, and evaluation is released at `https://github.com/CapitalOne-Research/con-rag`. All datasets used are public, and prompts and hyperparameters are given in Section 3 and the Appendix.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Experimental details are in Appendix Section A.7

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [N/A]

   Justification: Does not apply to our experimental setting.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Details in Experimental section and Appendix.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification:

   Guidelines: Conforms with the NeurIPS Code of Ethics to the best of our knowledge.

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: In Discussion section and Impact Statement.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: Not applicable to this work.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Assets are properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: Paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Not applicable to this work.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Not applicable to this work.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: LLMs used for evaluation.

Guidelines:

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