Teaching Large Language Models to Maintain Contextual Faithfulness via Synthetic Tasks and Reinforcement Learning

Anonymous ACL submission

Abstract

Teaching large language models (LLMs) to be faithful in the provided context is crucial for building reliable information-seeking systems. Therefore, we propose a systematic framework, CANOE, to improve the faithfulness of LLMs in both short-form and long-form generation tasks without human annotations. Specifically, we first synthesize short-form question-answering (QA) data with four diverse tasks to construct high-quality and easily verifiable training data without human annotation. Also, we propose Dual-GRPO, a rule-based reinforcement learning method that includes three tailored rule-based rewards derived from synthesized short-form QA data, while simultaneously optimizing both short-form and long-form response generation. Notably, Dual-GRPO eliminates the need to manually label preference data to train reward models and avoids over-optimizing short-form generation when relying only on the synthesized short-form QA data. Experimental results show that CANOE greatly improves the faithfulness of LLMs across 11 different downstream tasks, even outperforming the most advanced LLMs, e.g., GPT-4o and OpenAI o1.

1 Introduction

Recent progress in large language models (LLMs) has revolutionized text generation with their remarkable capabilities (OpenAI, 2023; DeepSeek-AI et al., 2025b). In practice, LLMs are widely used to generate fluent and coherent text responses based on the provided contextual information, e.g., document question answering (QA) (Wang et al., 2024) and text summarization (Zhang et al., 2024). However, LLMs often generate responses that are not faithful or grounded in the input context, i.e., faithfulness hallucinations (Ji et al., 2023; Huang et al., 2024), which can undermine their trustworthiness. Maintaining faithfulness to the context is especially important in fields where accurate information transfer is essential (Duong et al., 2025).

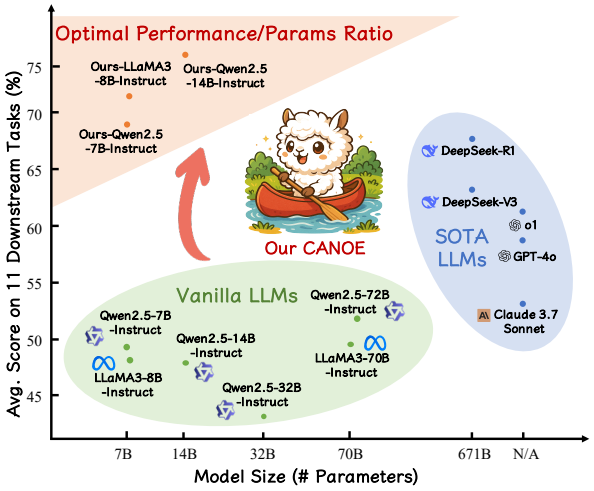

Figure 1: Average score on 11 downstream tasks vs model size. With only 7B parameters, CANOE already exceeds state-of-the-art LLMs like GPT-4o and o1.

For instance, in legal summarization (Dong et al., 2025), the text output must reflect the content of legal documents without introducing any distortions.

However, improving the faithfulness of LLMs faces three key challenges. Specifically, (1) **Faithfulness is difficult to improve by simply scaling model parameters**: Previous works (Xie et al., 2024; Li et al., 2025) find that LLMs may overly rely on internal knowledge learned from extensive pre-training data while disregarding provided contexts, i.e., the knowledge conflicts (Xu et al., 2024b). When the model parameters increase and internal knowledge grows, this may lead to greater knowledge conflicts and further lower the faithfulness of LLMs (Ming et al., 2025). Thus, it is necessary to explore the tailored post-training method to improve the faithfulness instead of simply scaling the model parameters. (2) **Faithfulness is challenging to consistently boost across different downstream tasks**: Recently, several methods (Li et al., 2024; Duong et al., 2025) have been proposed to improve the faithfulness of LLMs

for different tasks. For example, Bi et al. (2024) aligns LLMs through DPO (Rafailov et al., 2023) with constructed faithful and unfaithful short-form completions, improving the performance of LLMs on short-form QA tasks. However, these recent methods are designed for specific tasks, so they fail to consistently improve the faithfulness of LLMs across various tasks, like text summarization and multiple-choice questions, because these tasks can vary greatly. (3) **Data used to enhance faithfulness is hard to scale**: This issue is especially problematic with data used to improve the faithfulness in long-form generation tasks. Unlike tasks with clear answers, e.g., short-form fact-seeking QA tasks (Wei et al., 2024), there is no standard way to ensure data quality in long-form generation tasks (Duong et al., 2025). Thus, data is typically annotated by humans (Kryscinski et al., 2020; Zhu et al., 2023), which is costly and not scalable.

To tackle these challenges, we propose a systematic post-training method called CANOE. The main idea behind CANOE is to synthesize easily verifiable short-form QA data and then leverage reinforcement learning (RL) with tailored rule-based rewards to improve the faithfulness of LLMs in both short-form and long-form generation tasks. CANOE firstly introduces Dual-GRPO, a variant of GRPO (Shao et al., 2024) that includes three carefully tailored rule-based RL rewards derived from synthesized short-form QA data, while optimizing both short-form and long-form response generation. For the provided contextual information and question, Dual-GRPO first prompts LLMs to produce a reasoning process, followed by a long-form answer composed of detailed and complete sentences, and finally a concise short-form answer in just a few words. In this way, we can assign different rewards to long-form and short-form responses, optimizing both simultaneously. Note that we assign accuracy rewards on generated short-form responses since the short-form QA task enables reliable rule-based verification of faithfulness. To overcome the problem of the faithfulness of the generated long-form responses being difficult to evaluate via rule-based verification (Zheng et al., 2025; OpenAI, 2025), we propose proxy rewards to evaluate it implicitly. Specifically, we construct the new input by replacing the given context with the generated long-form answer, then feed it to the LLMs to evaluate whether a long-form answer can drive the LLMs toward the correct short-form answer. If the generated long-form response enables LLMs to generate the correct final answer, this indicates that it remains context-faithful and contains easy-to-understand sentences that answer the question correctly. We also introduce format rewards to ensure more structured outputs and contribute to more stable training. To obtain the data used for training without human annotation, we collect head-relation-tail triples from the knowledge base, apply the advanced GPT-4o (OpenAI, 2023) to synthesize the question and contextual information, and use the tail entity from the triple as the answer to ensure the correctness. Moreover, we introduce four diverse QA tasks to ensure the complexity and diversity of the training data. Combined with the rule-based Dual-GRPO and data synthesis, CANOE can teach LLMs to remain context-faithful in both short-form and long-form generation tasks without relying on human annotations.

We evaluate the effectiveness of CANOE across 11 different downstream tasks, covering short-form and long-form generation tasks. Results show that CANOE significantly reduces faithfulness hallucinations. Specifically, CANOE significantly improves the overall score, e.g., 22.6% for *Llama3-Instruct-8B*. Meanwhile, CANOE surpasses the most advanced LLMs (e.g., GPT-4o) in the overall score. To the best of our knowledge, these results are unprecedented for open-source models that do not rely on additional human annotations.

2 Related Work

Recently, the demand for utilizing LLMs to generate coherent text responses based on the provided contexts has continued to grow, particularly in text summarization and retrieval-augmented generation (RAG) scenarios. However, LLMs are often criticized for generating outputs that deviate from the provided contents, namely *faithfulness hallucination* (Li et al., 2022; Ji et al., 2023; Huang et al., 2024; Si et al., 2025). Many approaches have been proposed to improve the faithfulness of LLMs. The first line of work focuses on the inference stage of LLMs, such as designing prompts to encourage context integration (Zhou et al., 2023), improving context quality via explicit denoising (Xu et al., 2024a), and context-aware decoding to amplify contextual information (Shi et al., 2024). Although effective, these approaches primarily serve as a compensatory way rather than enabling the model to inherently learn to prevent generating unfaithful responses. Therefore, many studies attempt to

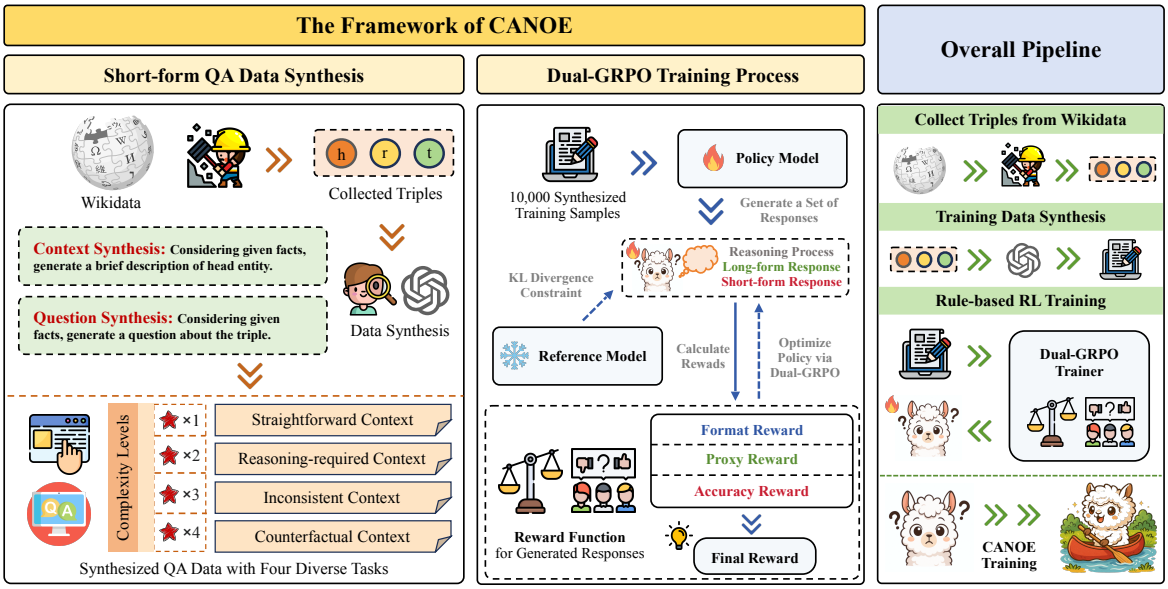

Figure 2: An overview of CANOE framework. CANOE first synthesizes easily verifiable short-form QA data and then proposes the Dual-GRPO with designed rule-based rewards to improve the faithfulness of LLMs.

apply post-training methods to improve the faithfulness. Bi et al. (2024) utilizes constructed faithful and unfaithful short-form completions and applies DPO to align LLMs to be context-faithful in short-form QA tasks. Huang et al. (2025) trains LLMs to discriminate between faithful and unfaithful responses in long-form QA tasks by unfaithful response synthesis and contrastive tuning. Duong et al. (2025) proposes a pipeline to generate a self-supervised task-specific dataset and applies preference training to enhance the faithfulness for a special task. However, these methods struggle to consistently improve the faithfulness of LLMs across various tasks, as these methods are designed for specific tasks. Thus, how to consistently improve the faithfulness of LLMs on different downstream tasks, including short-form and long-form generation tasks, still remains under-explored.

3 Methodology

In this section, we will detail our proposed framework CANOE, which aims to teach LLMs to remain faithful across different tasks without human annotation. Specifically, we first synthesize easily verifiable short-form QA data and then propose the Dual-GRPO with designed rule-based rewards to improve the faithfulness of LLMs in both short-form and long-form response generation. We start with the introduction of the short-form data synthesis process, then a brief overview of RL protocol, and the tailored rule-based rewards used in the proposed Dual-GRPO training. An overview of the CANOE framework is presented in Figure 2.

3.1 Training Data Construction

Constructing high-quality and easily verifiable data is crucial for rule-based RL training (Shao et al., 2024). Inspired by knowledge base question generation (Cui et al., 2019; Guo et al., 2024), we attempt to collect triples from the knowledge base and use the advanced LLMs to synthesize the context and question. Concretely, we first collect about 30,000 head-relation-tail triples from Wikidata (Vrandečić and Krötzsch, 2014). Each collected triple (h, r, t) includes a head entity h, a tail entity t, and the relation r between two entities. Then we craft prompt templates and query the most advanced GPT-4o to synthesize the contextual information c and question q based on the triple (h, r, t). We directly use the tail entity t as the final answer a to ensure the correctness and easy validation of the synthesized data. Each synthetic short-form QA sample (c, q, a) consists of a contextual passage c, a question q, and a ground truth answer a. In this way, we can obtain short-form QA data that can be easily verified, thus we can utilize a rule-based RL method to optimize our LLMs to be more faithful. Meanwhile, to ensure the complexity and diversity of training data, we design four diverse QA tasks, including straightforward context, reasoning-required context, inconsistent context, and counterfactual context. The model is

expected to answer the question by leveraging the information in the provided context.

Straightforward Context. A straightforward context means that the context clearly contains statements of the final answer. It requires models to accurately locate and utilize information from the context in order to answer questions. Specifically, we keep the original collected triple as input to query GPT-4o to synthesize the data (c, q, a).

Reasoning-required Context. This context contains multiple related entities and relations, and requires models to answer multi-hop reasoning questions. Firstly, we construct a subgraph based on the sampled triples and extract 2, 3, 4-hop paths $[(h^1, r^1, t^1), ..., (h^n, r^n, t^n)]_{n \leqslant 4}$. Then, we use the n-th tail entity t^n as the ground truth answer and employ the constructed paths to query GPT-4o to obtain the multi-hop context and question.

Inconsistent Context. This involves multiple randomly ordered contexts generated from different triples. This simulates noisy and inconsistent scenarios, where models need to detect inconsistencies and focus on useful and relevant contexts to answer the questions. We construct such a sample by combining the contexts from up to three QA samples.

Counterfactual Context. A counterfactual context contains statements that contradict common sense within the collected triples. Firstly, we replace the tail entity t of the original collected triple with a similar but counterfactual entity t^{cf}. Then, we query GPT-4o to generate questions and counterfactual contexts to construct counterfactual samples. Unlike the aforementioned tasks, this task further highlights the importance of faithfulness for LLMs to answer the questions correctly, as it prevents models from depending on their learned factual knowledge to find the right answers.

By introducing four different tasks, we construct 10,000 QA pairs used for training without human annotation. These short-form QA data can be easily verified and include tasks varying in complexity, which can make rule-based RL training more efficient in improving the faithfulness of LLMs. More details can be found in the Appendix A, e.g., used prompts, data mixing recipes, and data statistics.

3.2 Reinforcement Learning Protocol

For RL training of LLMs, methods based on policy optimization, such as PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024), have been explored. Given the effectiveness of GRPO in training models and its advantages over PPO, e.g., eliminating the need for human-annotated preference data to train a reward model, we utilize GRPO to optimize and improve the faithfulness of the policy model π_θ.

For each input, consisting of provided contextual information c, a natural language question q, the model generates a group of G candidate answers, $\{o_1, o_2, \ldots, o_G\}$. Each candidate is evaluated using a designed composite rule-based reward function to capture the end goal of faithfulness. GRPO leverages the relative performance of candidates within the group to compute an advantage A_i for each output, guiding policy updates according to the following objective:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{c,q,\{o_i\}\sim\pi_{\theta_{old}}} \left[\frac{1}{G} \sum_{i=1}^{G} \mathcal{L}_i - \beta \mathbb{D}_{KL}(\pi_\theta || \pi_{ref}) \right], \quad (1)$$

$$\mathcal{L}_i = \min \left(w_i A_i, \text{clip}(w_i, 1-\epsilon, 1+\epsilon) A_i \right), \quad (2)$$

where $w_i = \frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}$, $\pi_{\theta_{\text{old}}}$ is the policy before the update, π_{ref} is the reference policy (i.e., the initial model), ϵ and β are hyperparameters controlling the update step and divergence regularization and A_i is computed using the normalized reward within the group. We use synthesized short-form QA data as training data, which is easily verifiable, so that we can apply GRPO and train LLMs using the rule-based reward function. By generating multiple candidates per input, GRPO naturally accommodates the inherent challenges of utilizing the contextual information c and answering the question q, e.g., LLMs may overly rely on the internal knowledge while disregarding provided contexts. Meanwhile, employing the rule-based GRPO removes the need for humans to annotate short-form and long-form preference data used for training the reward model.

3.3 Reward Design

Having a well-designed reward is key to the effectiveness of RL training (Du et al., 2025). To use easily verifiable short-form QA data to improve the faithfulness, the most intuitive reward would be the accuracy reward, which can check if the generated responses match the ground truth answers. However, in our early experiments, we found that relying solely on short-form QA data and accuracy rewards fails to enhance the faithfulness of long-form response generation, as the models may over-optimize short-form generation and learn a false pattern. For example, the tuned models tend to simply copy text spans from the context as answers and lose their ability to generate long-form responses. Unfortunately, directly evaluating the

faithfulness of long, free-form responses via the rule-based verification continues to pose a significant and unresolved challenge.

Therefore, we propose **Dual-GRPO**, which includes a set of well-designed rewards that provide more harmonized guidance for optimizing LLMs to generate faithful responses. Unlike the original GRPO that over-optimizes short-form generation, we first prompt LLMs to generate both long-form and short-form responses, then assign different rewards to the two generated responses to improve the faithfulness of the two types of generation.

System Prompt and Rollouts. For the provided context and question, Dual-GRPO employs the designed system prompt that requires LLMs to produce a reasoning process, then a long-form answer composed of detailed and complete sentences, and finally a concise short-form answer in just a few words. For example, given the context, if the question is *"What is the country of origin of Super Mario?"*, the long answer could be *"Super Mario originated from Japan."*, while the short answer could simply be *"Japan"*. In this way, we can assign different reward scores to long-form and short-form answers while optimizing them both at once. This system prompt also triggers zero-shot chain-of-thought reasoning in the policy model, which progressively improves as training advances to optimize for the reward. The system prompt used for Dual-GRPO rollouts is shown in the Appendix B.

Accuracy Reward for Short-form Response Generation. This reward directly assesses whether the generated short-form responses match the ground truth answers. We use the exact matching (EM) to measure accuracy, giving a score of 1 for a match and 0 for a mismatch. Thus, we can ensure that the generated short-form response correctly answers the question based on the context, making LLMs more faithful in short-form response generation.

Proxy Reward for Long-form Response Generation. Evaluating the faithfulness of the generated long-form responses via the rule-based verification remains challenging. This is because these long-form answers are often free-form, making rule-based verification ineffective (Zheng et al., 2025; OpenAI, 2025). Therefore, instead of directly evaluating the faithfulness of the long-form response, we propose a proxy reward to evaluate it implicitly, as the faithfulness of a long-form answer can be measured by its ability to drive the LLMs toward a correct short-form answer. Specifically, for each generated long-form answer y_{lf}, we replace the given context c with it as new input and feed it to the LLM to check whether the LLM can produce the correct short-form answer based on y_{lf}. If the generated long-form response can enable the LLM to generate the correct answer, it indicates that the long-form response stays faithful to the context, contains complete and easy-to-understand sentences, and correctly addresses the question. Thus, we assign a reward score of 1 for the positive long-form response that helps the LLM to produce the correct final answer, and a reward score of 0 for those that lead to an incorrect answer.

Format Reward. We also include a format reward that encourages adherence to a predefined output structure (e.g., using <think>, <long_answer>, and <short_answer> tags). Outputs that conform to this pattern receive a reward boost, thereby enhancing clarity and consistency. We use the string matching method to evaluate whether the generated responses adhere to the format, giving a score of 1 for a match and 0 for a mismatch.

Finally, we use the sum of these three rewards as the final composite reward. It enhances the efficacy of the rule-based RL training framework, guiding the model toward generating more faithful responses in both short-form and long-form tasks. More details are shown in the Appendix B.

4 Experiments

In this section, we conduct experiments and provide analyses to justify the effectiveness of CANOE.

4.1 Tasks and Datasets

To evaluate our method CANOE comprehensively, we select a range of downstream datasets, including short-form and long-form generation tasks.

Short-form Generation Tasks. For short-form generation tasks, we use two counterfactual QA datasets (ConFiQA (Bi et al., 2024) and CNQ (Longpre et al., 2021)), a multiple-choice questions dataset FaithEval (Ming et al., 2025), and a factual QA dataset FiQA (Bi et al., 2024) that is the factual version of ConFiQA. These datasets ensure the answers appear in the contexts to evaluate the faithfulness. We also evaluate our method on four open-domain QA datasets within the FollowRAG benchmark (Dong et al., 2024) to evaluate the abilities of LLMs in real-world RAG scenarios, including NaturalQA (Kwiatkowski et al., 2019b), TriviaQA (Joshi et al., 2017), HotpotQA (Yang et al., 2018), and WebQSP (Yih et al., 2016). In real-world RAG

Model	Short-form Generation Tasks									Long-form Generation Tasks			Avg. Score	
	ConFiQA		FiQA		CNQ		FaithEval	FollowRAG		XSum	WikiLarge	CLAPNQ		
	EM	Acc	EM	Acc	EM	Acc	Acc	EM	Acc	FS	FS	FS	Avg EM	Avg Acc
The state-of-the-art LLMs														
GPT-4o	31.5	42.7	66.8	79.6	43.4	55.9	47.5	42.2	57.8	80.7	88.1	70.3	58.8	65.3
GPT-4o mini	49.5	63.7	67.1	78.8	47.8	54.3	50.9	38.5	51.3	75.4	91.0	66.0	60.8	66.4
DeepSeek V3	49.5	58.6	67.0	76.5	54.6	67.3	51.0	37.7	55.2	82.8	85.6	71.0	62.4	68.5
Claude 3.7 Sonnet	26.0	36.0	56.4	72.2	41.4	65.0	45.6	36.3	53.7	78.3	81.7	68.3	54.3	62.6
OpenAI o1	49.0	57.9	78.0	89.7	29.5	39.1	52.0	40.5	57.0	81.0	88.1	68.0	60.8	66.6
DeepSeek R1	68.4	74.3	68.4	80.7	60.3	70.2	60.1	42.9	56.6	80.3	83.0	73.5	67.1	72.3
Claude 3.7 Sonnet-Thinking	27.1	38.7	59.5	76.7	42.1	67.0	57.0	38.8	55.3	79.0	81.4	72.2	57.1	65.9
LLaMA-3-Instruct Series														
LLaMA-3-Instruct-8B	49.2	58.2	11.4	59.3	37.8	45.2	52.0	31.1	44.8	64.2	77.1	58.5	47.7	57.4
LLaMA-3-Instruct-70B	38.1	54.5	9.1	66.8	54.2	65.0	50.9	38.7	45.7	72.0	77.4	47.2	48.5	59.9
SFT-8B	65.1	70.3	35.9	59.9	52.6	65.7	43.0	19.2	21.0	62.2	74.2	55.3	50.9	56.4
Context-DPO-8B	66.3	72.9	40.9	59.5	54.6	62.3	37.5	29.9	43.8	65.2	78.2	59.1	54.0	59.8
SCOPE$_{sum}$-8B	35.7	64.6	7.1	68.7	33.8	60.6	55.7	30.1	46.2	70.3	80.3	59.8	46.6	63.3
CANOE-LLaMA-8B	**73.5**	**80.9**	**82.7**	**84.9**	**66.7**	**73.4**	**74.6**	**40.9**	**51.7**	**74.4**	**84.4**	**64.9**	**70.3**	**73.6**
Δ Compared to Vanilla.	+24.3	+22.6	+71.3	+25.6	+28.9	+28.2	+22.6	+9.8	+6.9	+10.2	+7.3	+6.4	+22.6	+16.2
Qwen-2.5-Instruct Series														
Qwen-2.5-Instruct-7B	52.5	61.0	13.2	68.4	55.3	68.2	56.1	32.6	45.3	63.4	57.8	61.2	49.0	60.2
Qwen-2.5-Instruct-14B	34.1	47.3	0.8	61.4	43.1	64.3	51.6	34.8	51.2	68.2	82.3	63.4	47.3	61.2
Qwen-2.5-Instruct-32B	44.5	66.4	39.2	81.1	37.7	66.4	47.0	33.9	53.1	20.2	57.7	31.7	39.0	52.9
Qwen-2.5-Instruct-72B	43.7	52.3	4.8	67.3	51.8	62.2	45.2	38.5	55.7	71.2	90.4	64.8	51.3	63.6
SFT-7B	62.8	69.8	48.8	76.6	60.1	65.3	50.3	29.0	41.7	55.2	51.3	57.2	51.8	58.4
Context-DPO-7B	64.5	70.6	57.1	78.2	62.3	70.1	45.7	31.0	43.7	60.2	53.4	62.8	54.6	60.6
SCOPE$_{sum}$-7B	39.3	47.9	12.9	60.9	50.2	55.3	52.3	30.6	46.0	68.3	72.0	63.2	48.6	58.2
CANOE-Qwen-7B	**67.6**	**75.2**	**78.1**	**83.5**	**67.2**	**76.4**	**70.5**	**37.0**	**50.2**	**72.4**	**86.1**	**65.2**	**68.0**	**72.4**
Δ Compared to Vanilla.	+15.1	+14.2	+64.9	+15.0	+11.9	+8.2	+14.4	+4.4	+4.9	+9.0	+28.3	+4.0	+19.0	+12.3
CANOE-Qwen-14B	**85.7**	**87.4**	**87.8**	**88.5**	**81.8**	**84.2**	**67.4**	**46.1**	**54.6**	**75.7**	**91.1**	**68.4**	**75.5**	**77.2**
Δ Compared to Vanilla.	+51.6	+40.1	+87.0	+27.1	+38.7	+19.9	+15.8	+11.3	+3.4	+7.5	+8.8	+5.0	+28.2	+16.0

Table 1: Experimental results (%) on eleven datasets. The FollowRAG results represent the results averaged over these four open-domain QA datasets as shown in Table 7, including NaturalQA, TriviaQA, HotpotQA, and WebQSP. **Bold** numbers indicate the best performance of models with the same model size. Avg EM/Acc represents the average score between short-form task metrics (EM/Acc) and long-form task metric FaithScore (FS).

scenarios, the answer may not appear in the retrieved passages, and these passages tend to be noisy. We evaluate models based on whether gold answers are included in the generated responses (i.e., Acc) following Asai et al. (2024) and exact matching (EM) for QA tasks. For multiple-choice questions, we follow Ming et al. (2025) and use keyword matching to verify the accuracy.

Long-form Generation Tasks. We include a text summarization task XSum (Narayan et al., 2018), a text simplification task WikiLarge (Zhang and Lapata, 2017), and a long-form QA task CLAPNQ (Rosenthal et al., 2025). To evaluate the faithfulness of generated long-form answers, called FaithScore (FS), we use MiniCheck (Tang et al., 2024) to check whether the model response is grounded in the provided context. MiniCheck is a state-of-the-art method to recognize if LLM output can be grounded in given contexts. If the model response contains at least one statement that cannot be inferred from the context, we consider it as a negative response; otherwise, it is a positive response. We also query GPT-4o to evaluate the quality of generated responses, namely QualityScore.

More details are available in the Appendix C.

4.2 Baselines and Implementation Details

Baselines. We compare several baselines, including (1) Vanilla LLMs: including LLaMA-3-Instruct (Grattafiori et al., 2024) and Qwen-2.5-Instruct (Yang et al., 2024) of different sizes. We also conduct supervised fine-tuning on synthesized 10,000 short-form data as SFT baselines; (2) SOTA LLMs: We further evaluate the most advanced LLMs, including GPT-4o, GPT-4o-mini, OpenAI o1 (Jaech et al., 2024), Claude 3.7 Sonnet (Anthropic, 2025), Claude 3.7 Sonnet-Thinking, Deepseek R1, and Deepseek V3 (DeepSeek-AI et al., 2025a,b); (3) The Designed Methods to Improve Faithfulness of LLMs: Context-DPO (Bi et al., 2024) aligns LLMs through DPO with constructed faithful and unfaithful short-form answers, thus improving the faithfulness in short-form generation. SCOPE (Duong et al., 2025) introduces a pipeline to generate self-supervised task-specific data and applies preference training to enhance the faithfulness in a special task. We train it on the sampled training set of the summarization task XSum as SCOPE$_{sum}$, regarding it as the method designed to improve the faithfulness of long-form response generation.

Implementation Details. Our main experiments are conducted on LLaMA-3-Instruct and Qwen-2.5-Instruct. More implementation details are shown in Appendix D, e.g., hyperparameters.

4.3 Main Results

CANOE Improves the Faithfulness of LLMs in Both Short-form and Long-form Response Gen-

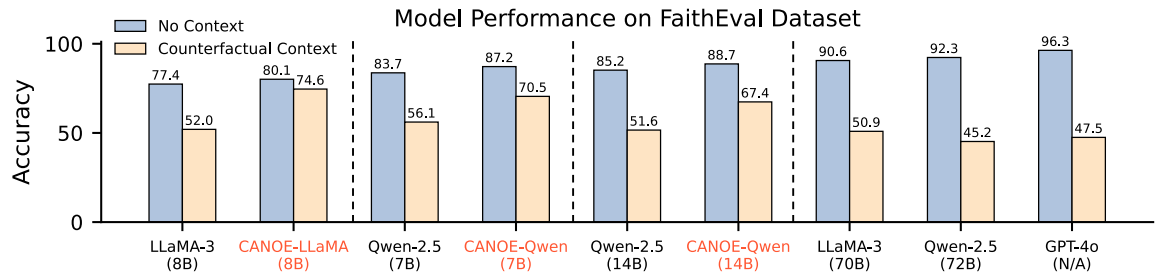

Figure 3: Model performance comparison on FaithEval in a closed-book QA setting and counterfactual context setting. Our models are colored in orange. We report the results from the chat version of LLaMA-3 and Qwen-2.5.

Model	XSum	WikiLarge	CLAPNQ	Avg
GPT-4o	98.5	97.5	81.2	92.4
LLaMA-3-Instruct-8B	70.9	82.9	39.2	64.3
LLaMA-3- Instruct-70B	86.2	83.0	30.1	66.4
CANOE-LLaMA-8B	**85.8**	**87.8**	**65.5**	**79.7**
Qwen-2.5-Instruct-7B	79.4	79.0	64.6	74.3
Qwen-2.5-Instruct-14B	90.5	83.1	63.6	79.1
Qwen-2.5-Instruct-32B	90.3	83.9	58.6	77.6
Qwen-2.5-Instruct-72B	95.7	94.1	75.4	88.4
CANOE-Qwen-7B	**91.5**	**87.3**	**68.2**	**82.3**
CANOE-Qwen-14B	**91.9**	**89.7**	**73.5**	**85.0**

Table 2: QualityScore on long-form generation tasks.

Model	Acc			EM		
	QA	MR	MC	QA	MR	MC
GPT-4o	52.2	45.6	30.3	43.3	32.4	18.7
LLaMA-3-Instruct-8B	69.7	55.9	49.1	60.0	47.9	39.6
CANOE-LLaMA-8B	**82.7**	**80.1**	**79.8**	**76.4**	**73.5**	**70.5**
Qwen-2.5-Instruct-7B	72.8	59.1	51.1	64.9	50.2	42.5
Qwen-2.5-Instruct-14B	62.4	44.9	34.7	44.7	34.3	23.3
Qwen-2.5-Instruct-32B	74.1	65.9	59.3	55.9	42.8	34.8
Qwen-2.5-Instruct-72B	63.3	50.3	43.3	54.3	42.2	34.7
CANOE-Qwen-7B	**79.5**	**76.1**	**70.1**	**73.3**	**67.9**	**61.7**
CANOE-Qwen-14B	**91.8**	**86.4**	**84.1**	**89.7**	**85.2**	**82.1**

Table 3: Results (%) on three tasks in ConFiQA.

eration. As shown in Table 1, CANOE shows consistent and significant improvements on 11 datasets measuring faithfulness. CANOE achieves substantial improvements in the overall score compared to original LLMs, e.g., **22.6%** for *Llama3-8B* and **19.0%** for *Qwen2.5-7B* in Avg EM score. CANOE also surpasses the most advanced LLMs (e.g., GPT-4o) in the overall score (both Avg EM and Avg Acc scores). This shows that CANOE can effectively align LLMs to be context-faithful. Meanwhile, for real-world RAG scenarios, our proposed CANOE can also improve the performance even though the answer may not appear in the retrieved passages, and these passages are often noisy.

CANOE Maintains the Factuality of LLMs. We further evaluate whether CANOE will reduce the factuality of LLMs. Following Ming et al. (2025), we modify the original FaithEval and make it a closed-book QA setting, where no context is provided and LLMs need to give factual answers. In this case, the models rely entirely on their parametric knowledge of common facts, and we find that our proposed CANOE maintains the factuality compared to the untuned LLM as shown in Figure 3. However, when a new context with counterfactual evidence that contradicts the model's parametric knowledge is introduced, performance declines sharply. For example, GPT-4o achieves 96.3% accuracy on factual closed-book QA task but only 47.5% on counterfactual QA task that evaluates the faithfulness of LLMs. This highlights that, unlike factuality, the faithfulness of LLMs is difficult to improve by simply scaling model parameters, which further indicates the necessity of a post-training method to improve faithfulness.

CANOE Improves the Quality of Long-form Response Generation. As shown in Table 2, we can find that our proposed CANOE also improves the quality of generations. This is because the proxy reward implicitly requires LLMs to generate easy-to-understand responses, which further optimizes the response quality. CANOE consistently improves the generation quality in the three long-form tasks, which illustrates the effectiveness of our method.

CANOE Enhances LLMs' Reasoning in Short-form Response Generation. ConFiQA consists of three different tasks: question answering (QA), multi-hop reasoning (MR), and multi-conflicts reasoning (MC). QA focuses on the single-hop task with context containing one corresponding answer, while MR and MC involve multi-hop reasoning tasks with context containing one and multiple related counterfactual contexts, respectively. As shown in Table 3, CANOE not only improves the faithfulness in the single-hop QA task but also enhances the reasoning ability in reasoning tasks.

CANOE Mitigates Overconfidence Bias. For each model, we select a total of 110 unfaithful samples with the highest perplexity from the 11 datasets, 10 samples per dataset. Then we report the average

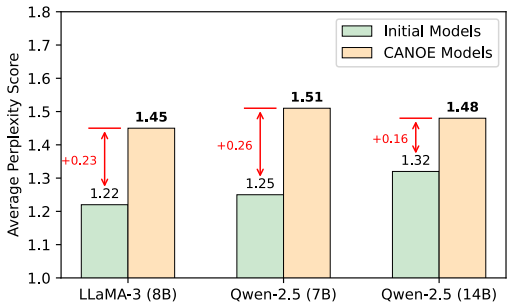

Figure 4: The average perplexity score of 110 negative samples for each model from eleven datasets.

Model	Short-form Tasks		Long-form Tasks	
	EM	Acc	FaithScore	QualityScore
CANOE-LLaMA-8B	**67.7**	**73.1**	**74.6**	**79.7**
-w/o. Dual-GRPO & Data Synthesis	36.3	51.9	66.6	64.3
-w/o. Dual-GRPO (i.e., original GRPO)	60.5	66.6	N/A	23.5
-w/o. Reasoning-required Context.	63.7	69.4	71.7	75.3
-w/o. Inconsistent Context.	64.4	70.2	70.2	72.5
-w/o. Counterfactual Context.	62.6	67.8	69.7	73.7

Table 4: Results of ablation study. EM/Acc in short-form tasks represents the average score between QA metrics (EM/Acc) and the accuracy of FaithEval. N/A means a false generation pattern hacks this metric.

perplexity score on these negative samples shown in Figure 4. We can find that CANOE produces the high perplexity scores, indicating low confidence scores, for these bad cases. This shows that CANOE mitigates overconfidence in these false statements.

4.4 Analysis

Ablation Study. We conduct an ablation study in Table 4. The result reveals that our proposed CANOE (including Dual-GRPO and the designed short-form data synthesis) significantly improves the faithfulness of LLMs in both short-form and long-form generation. For Dual-GRPO, we observe that directly applying GRPO to synthesized short-form data leads to over-optimizing short-form generation and a false response generation pattern. We find that tuned models tend to directly copy text spans from the given context as the final answer instead of following instructions in long-form generation tasks (we show the case study in the Appendix F.5 to visually show this phenomenon). Thus, the generated responses do not contain syntactically and semantically complete sentences for long-form generation tasks, which leads to low QualityScore performance and also invalidates the metric used for evaluating faithfulness. We also explore the effectiveness of our training data construction strategy. For the designed QA tasks used to ensure the complexity and diversity of training data, we replace the designed Reasoning-required Context and Inconsistent Context samples with the same number of samples that contain Straightforward Context. We find that involving these more challenging instances can improve the effectiveness of RL training. We also replace the data points that contain Counterfactual Context with the same number of factual samples. The designed Counterfactual Context improves the final performance as it prevents models from depending on their learned factual knowledge to find the right answers.

Question: What do Greeks say when they break plates?
Qwen-2.5-Instruct-7B: Based on the information provided in the passages, there is no specific mention of what exactly Greeks say when they break plates during the plate-breaking ceremony. (✗)
CANOE-Qwen-7B: According to the information given, in the Three's Company episode "Opening Night," when Jack breaks the plates at the end of the celebration, he exclaims "Opa!".(✓) This suggests that "Opa!" might be a phrase commonly used in Greek culture during such festivities. (✓) "Opa!" is often used to express joy, excitement, or approval, which aligns with the celebratory nature of plate smashing in Greek traditions. (✓)

Table 5: Case study from long-form QA task CLAPNQ. For different useful statements, we use different colors.

Case Study. We further conduct a case study in Table 5 to visually show the advantages of CANOE. Our method ensures the statements are faithful and comprehensive, and the text flows naturally.

Human Evaluation. Evaluating long-form generation tasks remains challenging (Li et al., 2024). Thus, we conduct human evaluation in the Appendix E to show the effectiveness of our method.

Discussion. We also discuss some possible concerns about CANOE in the Appendix F, e.g., the effect of the amount of synthesized data.

5 Conclusion

In this paper, we propose CANOE, a systematic post-training method for teaching LLMs to remain faithful in both short-form and long-form generation tasks without human annotations. By synthesizing diverse short-form QA data and introducing Dual-GRPO, a tailored RL method with three well-designed rule-based rewards, CANOE effectively improves the faithfulness of LLMs. We first synthesize short-form QA data with four diverse tasks to construct high-quality and easily verifiable training data without human annotation. We then propose Dual-GRPO, a rule-based RL method that includes three tailored rule-based rewards derived from synthesized short-form QA data, while optimizing both short-form and long-form response generation simultaneously. Experimental results show that CANOE consistently improves the faithfulness of LLMs across diverse downstream tasks.

Limitations

Although experiments have confirmed the effectiveness of the proposed CANOE, four major limitations remain. Firstly, CANOE synthesizes short-form QA data and uses the proposed Dual-GRPO to improve the faithfulness of LLMs in long-form response generation implicitly; thus, how to directly synthesize long-form data and improve the faithfulness remains under-explored. Meanwhile, the synthesized short-form QA data is single-turn; thus, exploring the synthesis of multi-turn QA data presents an attractive direction for future research. The motivation behind our work is to improve the faithfulness of LLMs without human annotation, but it is still worth exploring how to incorporate the existing manually labeled data to further improve the faithfulness of the model. Finally, while our method achieves strong results, exploring additional strategies, e.g., using cold-start to get a better initial policy model and improve the reward scores in training for better performance across different downstream tasks is also a promising direction.

References

Anthropic. 2025. Claude 3.7 sonnet system card.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Baolong Bi, Shaohan Huang, Yiwei Wang, Tianchi Yang, Zihan Zhang, Haizhen Huang, Lingrui Mei, Junfeng Fang, Zehao Li, Furu Wei, Weiwei Deng, Feng Sun, Qi Zhang, and Shenghua Liu. 2024. Context-dpo: Aligning language models for context-faithfulness. *Preprint*, arXiv:2412.15280.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Wen Cui, Minghui Zhou, Rongwen Zhao, and Narges Norouzi. 2019. KB-NLG: From knowledge base to natural language generation. In *Proceedings of the 2019 Workshop on Widening NLP*, pages 80–82, Florence, Italy. Association for Computational Linguistics.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang,

Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. 2025b. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Guanting Dong, Xiaoshuai Song, Yutao Zhu, Runqi Qiao, Zhicheng Dou, and Ji-Rong Wen. 2024. Toward general instruction-following alignment for retrieval-augmented generation. *Preprint*, arXiv:2410.09584.

Xiangyun Dong, Wei Li, Yuquan Le, Zhangyue Jiang, Junxi Zhong, and Zhong Wang. 2025. TermDiffuSum: A term-guided diffusion model for extractive summarization of legal documents. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3222–3235, Abu Dhabi, UAE. Association for Computational Linguistics.

Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze Li, Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su, Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye, Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong, Shaowei Liu, Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong, Weiran He, Weixiao Huang, Wenhao Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu, Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping Bao, Yulun Du, Yuxin Wu, Yuzhi Wang, Zaida Zhou, Zhaoji Wang, Zhaowei Li, Zhen Zhu, Zheng Zhang, Zhexu Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, and Zonghan Yang. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *Preprint*, arXiv:2501.12599.

Song Duong, Florian Le Bronnec, Alexandre Allauzen, Vincent Guigue, Alberto Lumbreras, Laure Soulier, and Patrick Gallinari. 2025. SCOPE: A self-supervised framework for improving faithfulness in conditional text generation. In *The Thirteenth International Conference on Learning Representations*.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth

Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun

Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Shasha Guo, Jing Zhang, Xirui Ke, Cuiping Li, and Hong Chen. 2024. Diversifying question generation over knowledge base via external natural questions. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5096–5108, Torino, Italia. ELRA and ICCL.

Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, Xuan Liu, Tian Wu, and Haifeng Wang. 2018. DuReader: a Chinese machine reading comprehension dataset from real-world applications. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 37–46, Melbourne, Australia. Association for Computational Linguistics.

Lei Huang, Xiaocheng Feng, Weitao Ma, Yuchun Fan, Xiachong Feng, Yangfan Ye, Weihong Zhong, Yuxuan Gu, Baoxin Wang, Dayong Wu, Guoping Hu, and Bing Qin. 2025. Improving contextual faithfulness of large language models via retrieval heads-induced optimization. *Preprint*, arXiv:2501.13573.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2024. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.* Just Accepted.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019a. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019b. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Kun Li, Tianhua Zhang, Yunxiang Li, Hongyin Luo, Abdalla Moustafa, Xixin Wu, James Glass, and Helen Meng. 2025. Generate, discriminate, evolve: Enhancing context faithfulness via fine-grained sentence-level self-evolution. *Preprint*, arXiv:2503.01695.

Taiji Li, Zhi Li, and Yin Zhang. 2024. Improving faithfulness of large language models in summarization via sliding generation and self-consistency. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8804–8817, Torino, Italia. ELRA and ICCL.

Wei Li, Wenhao Wu, Moye Chen, Jiachen Liu, Xinyan Xiao, and Hua Wu. 2022. Faithfulness in natural language generation: A systematic survey of analysis, evaluation and optimization methods. *Preprint*, arXiv:2203.05227.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

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

made of marshmallows". In *The Thirteenth International Conference on Learning Representations*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

OpenAI. 2025. Deep research system card. Technical report, OpenAI.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Abhilasha Ravichander, Shrusti Ghela, David Wadden, and Yejin Choi. 2025. Halogen: Fantastic llm hallucinations and where to find them. *Preprint*, arXiv:2501.08292.

Sara Rosenthal, Avirup Sil, Radu Florian, and Salim Roukos. 2025. CLAPnq: Cohesive long-form answers from passages in natural questions for RAG systems. *Transactions of the Association for Computational Linguistics*, 13:53–72.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 783–791, Mexico City, Mexico. Association for Computational Linguistics.

Shuzheng Si, Haozhe Zhao, Gang Chen, Cheng Gao, Yuzhuo Bai, Zhitong Wang, Kaikai An, Kangyang Luo, Chen Qian, Fanchao Qi, Baobao Chang, and Maosong Sun. 2025. Aligning large language models to follow instructions and hallucinate less via effective data filtering. *Preprint*, arXiv:2502.07340.

Shuzheng Si, Haozhe Zhao, Gang Chen, Yunshui Li, Kangyang Luo, Chuancheng Lv, Kaikai An, Fanchao Qi, Baobao Chang, and Maosong Sun. 2024. Gateau: Selecting influential sample for long context alignment. *arXiv preprint arXiv:2410.15633*.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. Minicheck: Efficient fact-checking of llms on grounding documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Minzheng Wang, Longze Chen, Fu Cheng, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, Yunshui Li, Min Yang, Fei Huang, and Yongbin Li. 2024. Leave no document behind: Benchmarking long-context LLMs with extended multi-doc QA. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5627–5646, Miami, Florida, USA. Association for Computational Linguistics.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *Preprint*, arXiv:2411.04368.

Han Wu, Mingjie Zhan, Haochen Tan, Zhaohui Hou, Ding Liang, and Linqi Song. 2023. VCSUM: A versatile Chinese meeting summarization dataset. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6065–6079, Toronto, Canada. Association for Computational Linguistics.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *The Twelfth International Conference on Learning Representations*.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2024a. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *The Twelfth International Conference on Learning Representations*.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024b. Knowledge conflicts for LLMs: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8541–8565, Miami, Florida, USA. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng

Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Wentau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Annual Meeting of the Association for Computational Linguistics*.

Haopeng Zhang, Philip S. Yu, and Jiawei Zhang. 2024. A systematic survey of text summarization: From statistical methods to large language models. *Preprint*, arXiv:2406.11289.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594, Copenhagen, Denmark. Association for Computational Linguistics.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *Preprint*, arXiv:2504.03160.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. Context-faithful prompting for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14544–14556, Singapore. Association for Computational Linguistics.

Rongxin Zhu, Jianzhong Qi, and Jey Han Lau. 2023. Annotating and detecting fine-grained factual errors for dialogue summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6825–6845, Toronto, Canada. Association for Computational Linguistics.

Appendix

This appendix is organized as follows.

- In Section A, we report the details of constructing training data, e.g., the used triples and introduction of four designed tasks.
- In Section B, we go into detail about the proposed Dual-GRPO, including the system prompt and formal expressions of three different well-designed rewards.
- In Section C, we show the details of evaluations, e.g., the introduction of the used benchmarks and evaluation prompts.
- In Section D, we show the details of our implementation and training, e.g., hyperparameters and the used GPUs.
- In Section E, we show the implementation details of human evaluation.
- In Section F, we discuss some possible questions about the proposed CANOE. For example, we discuss the effect of the amount of synthesized short-form data for RL training.

A Training Data Details

A.1 Triples from Wikidata

To ensure the usability of the synthetic data and collected triples, we follow Bi et al. (2024) to collect entities corresponding to the top 1,000 most-visited Wikipedia pages from 2016 to 2023 and 41 relations selected by Bi et al. (2024) shown in Table 12. The most-visited Wikipedia pages are based on monthly page views and retain the most popular entities using criteria such as the number of hyperlinks. We finally collected 6,316 entities and 30,762 triples. We randomly select these triples to synthesize our training data, and finally construct 10,000 samples as the final training data.

A.2 Construction of Four Different Tasks

We design four different tasks to enhance the complexity and diversity of our training data. Meanwhile, we select *GPT-4o-2024-08-06* to construct the contexts and questions.

Straightforward Context. As shown in Sec. 3.1, we keep the original collected factual triple as input to query GPT-4o to synthesize the data (c, q, a). The prompts for querying GPT-4o to obtain the generated questions and contexts can be found in Figure 7 and Figure 8. We finally keep 2,000 such samples in the synthesized 10,000 training data, i.e., 20% of the data.

Reasoning-required Context. We construct paths $[(h^1, r^1, t^1), ..., (h^n, r^n, t^n)]_{n\leqslant 4}$ from a sub-graph; more details can be found in Sec. 3.1. Then, we use the n-th tail entity t^n as the ground truth answer and use the constructed paths to query GPT-4o to obtain the multi-hop context and question. The prompts for querying GPT-4o to obtain the generated questions and contexts can be found in Figure 9 and Figure 10. We finally keep 2,000 such samples in the synthesized 10,000 training data, i.e., 20% of the data.

Inconsistent Context. This involves multiple randomly ordered contexts generated from different triples. This simulates noisy and inconsistent scenarios, where models need to detect inconsistencies and focus on useful and relevant contexts to answer the questions. We construct such a sample by combining the contexts from up to three QA samples with reasoning-required context and use the original t^n as the answer. In this way, we can obtain more complex samples than ones with the reasoning-required context. To avoid duplicating the 2,000 samples with the reasoning-required context collected above, we reconstruct the new samples with the reasoning-required context used to obtain the samples with the inconsistent context. We keep 1,000 such samples in the synthesized 10,000 training data, i.e., 10% of the data.

Counterfactual Context. A counterfactual context includes statements that go against common sense found in the collected triples. Specifically, we construct samples with counterfactual contexts below by modifying previously collected triples (of three types, including straightforward context, reasoning-required context, and inconsistent context). We replace the tail entity t of the original collected triple with a similar but counterfactual entity t^{cf}, which is obtained by query GPT-4o using prompt *"Generate me a noun for an entity that is similar to the {t} but different, and require the entity to exist in the real-world, please tell me the answer directly:"*. Then, we query GPT-4o to generate questions and counterfactual contexts to construct counterfactual samples, using the counterfactual triples. The prompts used in constructing samples with counterfactual contexts are the same as the prompts used in constructing the three different tasks above. The reason we construct samples with counterfactual context in this way

Type	Num	Avg Len
Straightforward Context.	2,000	186.3
Reasoning-required Context.	2,000	262.2
Inconsistent Context.	1,000	421.2
Counterfactual Context.	5,000	260.8

Table 6: Statistics of the training data. Num indicates the number of samples. Avg Len shows the average length of the samples, including the context and question.

is that this prevents the model from learning the appropriate factual knowledge to answer the question correctly, rather than correctly exploiting the given contextual information. Therefore, we construct the same number of samples as the summed number of the three types above (including straightforward context, reasoning-required context, and inconsistent context), i.e., 5000 samples (50% of the data). Meanwhile, this task stresses the importance of keeping answers faithful in contexts, as it stops them from relying solely on the learned knowledge of LLMs to provide correct answers.

A.3 Statistics

We show the statistics of the training data in Table 6. Even though the length of the data we synthesize is short, we find that our model can be generalized with consistently state-of-the-art results on a wide range of tasks with different input lengths by utilizing our proposed Dual-GRPO, e.g., long-form QA and RAG generation with long texts as inputs.

B Dual-GRPO Details

In this section, we give a more detailed introduction to our proposed Dual-GRPO, including the designed system prompt and formal expressions of three different rewards.

System Prompt. For the provided contextual information and question, Dual-GRPO employs the designed system prompt that requires LLMs to produce a reasoning process, then a long-form answer that consists of detailed and complete sentences, and finally a concise short-form answer in just a few words. In this way, we can assign different reward scores to long-form answers and short-form answers while optimizing them both at once. Meanwhile, this system prompt also triggers zero-shot chain-of-thought reasoning in the policy model, which progressively improves as training advances to optimize for the reward. We use the same system prompt to train both LLaMA and Qwen models. We show our used system prompt in Figure 11.

Accuracy Reward. For short-form generation, we directly assign the accuracy reward. Specifically, for the generated short-form response y_{sf} based on the given context c and question q, which is extracted from the whole generated response y_{whole} via string matching, and the ground truth answer y_{gt} from the synthesized training data, the accuracy reward R_{acc} for the LLM θ can be calculated as:

$$R_{\text{acc}} = \begin{cases} 1 & \text{if } y_{sf}(c, q|\theta) = y_{gt}, \\ 0 & \text{otherwise.} \end{cases}$$

We use the exact matching (EM) to measure accuracy, giving a score of 1 for a match and 0 for a mismatch. In this way, we can ensure that the generated short-form response correctly answers the question based on the given context, making LLMs more faithful in short-form response generation.

Proxy Reward. Instead of directly evaluating the faithfulness of the generated long-form response, we propose a proxy reward to evaluate it implicitly. Specifically, for each generated long-form answer y_{lf}, we replace the given context c with it as new input and infer the LLM θ to determine whether the LLM can produce the correct short-form answer y_{sf} based on y_{lf} for the question q. Thus, the proxy reward R_{proxy} can be calculated as:

$$R_{\text{proxy}} = \begin{cases} 1 & \text{if } y_{sf}(y_{lf}, q|\theta) = y_{gt}, \\ 0 & \text{otherwise.} \end{cases}$$

If the generated long-form response can help LLMs generate the correct answer, it indicates that the long-form response is faithful to the context, contains syntactically and semantically complete sentences, and correctly addresses the question. Thus, we assign a reward score of 1 for the positive long-form response that helps the LLM to produce the correct answer, and a reward score of 0 for those that lead to incorrect answers.

Format Reward. To enforce the desired output format, we assign a reward on the whole generated response y_{whole} to evaluate whether it contains the proper XML tags. We use three types of tags as shown in our system prompt, as shown in Figure 11, including <think>, <long_answer>, and <short_answer> tags. Formally,

$$R_{\text{format}} = \begin{cases} 1 & \text{if correct formatting is present,} \\ 0 & \text{if incorrect formatting.} \end{cases}$$

We use the string matching method to evaluate whether the responses adhere to the format.

Final Reward. Finally, we use the sum of these three rewards as the final composite reward R_{final}. This well-designed reward R_{final} of Dual-GRPO enhances the efficacy of the rule-based RL training framework to guide the model toward generating more faithful responses in both short-form and long-form tasks. Formally,

$$R_{\text{final}} = R_{\text{acc}} + R_{\text{proxy}} + R_{\text{format}}.$$

Finally, we use this reward R_{final} to compute an advantage A_i for each output, guiding policy updates according to the GRPO objective.

Potential Reward Hacking Concerns. In the early experiments, we have also tried adding the length reward for long-form responses (i.e., the content between <long_answer> and </long_answer> tags) to avoid the potential reward hacking, e.g., avoiding the policy model directly copying the given context as the long-form response, but found that the task performance does not have a significant difference.

C Evaluation Details

C.1 Datasets

ConFiQA (Counterfactual QA). This is a dataset that incorporates knowledge conflicts through counterfactual passages to evaluate the faithfulness of LLMs on short-form generation. ConFiQA consists of three tasks: QA (Question Answering), MR (Multi-hop Reasoning), and MC (Multi-Conflicts). QA features single-hop question-answering tasks with context containing one corresponding counterfactual, while MR and MC involve multi-hop reasoning tasks with context containing one and multiple related counterfactual contexts, respectively. ConFiQA contains 1,500 data points used for testing (500/500/500 from QA/MC/MR).

CNQ (Counterfactual QA). CNQ is constructed based on Natural Questions (Kwiatkowski et al., 2019a). In CNQ, the context is modified to support counterfactual answers following (Longpre et al., 2021). It contains 2,773 samples that incorporate counterfactual passages to evaluate the faithfulness of LLMs on short-form generation.

FaithEval (Counterfactual Multiple-choice QA). FaithEval is a novel and comprehensive benchmark tailored to evaluate the faithfulness of LLMs in contextual scenarios across three diverse tasks: unanswerable, inconsistent, and counterfactual contexts. We select the counterfactual task to evaluate the faithfulness of LLMs, which contains 1,000 multiple-choice QA samples curated based on ARC-Challenge (Clark et al., 2018).

FiQA (Factual QA). FiQA is a factual version of ConFiQA, which shares the same questions as ConFiQA but contains the factual contexts and answers. The contexts and answers are provided by Bi et al. (2024), thus we can evaluate the faithfulness of LLMs in factual short-form response generation. It contains 1,500 samples for evaluation.

FollowRAG (RAG Scenarios for short-form QA). FollowRAG aims to assess the model's ability to follow user instructions in complex multi-document contexts. It consists of four well-known open-domain QA datasets for RAG scenarios, including NaturalQA, TriviaQA, HotpotQA, and WebQSP. We utilize the provided passages in FollowRAG as context and original query (instead of the version with added instruction constraints proposed by Dong et al. (2024)) as questions. We also use the original answers to report the results. FollowRAG contains 2,800 samples used for testing (700/700/700/700 from NaturalQA/TriviaQA/HotpotQA/WebQSP). Different from short-form generation tasks that the contexts always contain answers, in real-world RAG scenarios, the answer may not appear in the retrieved passages, and these passages tend to be noisy.

XSum (Summarization). Summarization is a content-grounded task where a model is provided a piece of text and tasked with synthesizing the most salient information within that text. XSum is a widely used dataset for text summarization, which consists of about 220,000 BBC articles as input documents. To facilitate our evaluation, we use the first 1,000 data points from the test set to evaluate our method.

WikiLarge (Simplification). Text simplification is a content-grounded task where a model is provided a piece of text and is tasked with paraphrasing it to make the text easier to read and understand. We use 1k instances sampled from the WikiLarge dataset as a test set, following Ravichander et al. (2025).

CLAPNQ (Long-form QA). CLAPNQ is a grounded long-form QA benchmark dataset for Retrieval Augmented Generation of LLMs. The answers are typically long, 2-3 sentences grounded on a single gold passage, in contrast to datasets based on machine reading comprehension, such as short-form Natural Questions, which are just a few words. CLAPNQ includes long answers with grounded gold passages from Natural Questions. We utilize the provided passages and ques-

tions from the dev set to evaluate the faithfulness of LLMs in long-form response generation for open-domain questions, which contains 600 data points.

C.2 Metrics and LLM-as-a-Judge

Metrics for Short-form Generation Tasks. We evaluate performance based on whether gold answers are included in the generated responses (i.e., Acc) following Asai et al. (2024) and exact matching (EM) for QA tasks. For multiple-choice questions in FaithEval, we use keyword matching to verify the accuracy, i.e., Acc.

Metrics for Long-form Generation Tasks. To evaluate the faithfulness of generated long-form answers, we use MiniCheck to check whether the model response is grounded in the provided context. MiniCheck is a state-of-the-art method to recognize if LLM output can be grounded in given contexts. We select the MiniCheck-FT5[1] because it is the best fact-checking model, outperforming GPT-4o in evaluating the faithfulness. If the model response contains at least one statement that cannot be inferred from the context, we consider it as a negative response; otherwise, it is a positive response. To evaluate the quality of the generated long-form responses for three different tasks (QualityScore), including summarization, simplification, and long-form QA, we design different prompts to query *GPT-4o-2024-11-20* as a judge to get the quality scores. We report the average results of the quality score results by querying GPT-4o twice. The prompts for three tasks can be found in Figure 12, Figure 13, and Figure 14.

C.3 Baselines

For SOTA LLMs, we select the following versions of these models to report the results. Specifically, we use *GPT-4o-2024-08-06* for GPT-4o, *GPT-4o-mini-2024-07-18* for GPT-4o-mini, *Claude 3.7 Sonnet-2025-02-19* for Claude 3.7 Sonnet and Claude 3.7 Sonnet-thinking, *Deepseek R1 2025-01-20* for Deepseek R1, *Deepseek V3 2024-12-26* for Deepseek V3, and *o1-2024-12-17* for OpenAI o1. To get stable experimental results, we query these models twice and report the average results on each task. For the methods that are designed for improving the faithfulness, we reproduce their released code based on *LLaMA-3-Instruct* and *Qwen-2.5-Instruct*. For SCOPE, we train it on the 10,000 sampled training set of the summarization task XSum as SCOPE$_{sum}$, which keeps the same number of data we used for training CANOE and provides a fair comparison.

[1]https://huggingface.co/lytang/MiniCheck-Flan-T5-Large

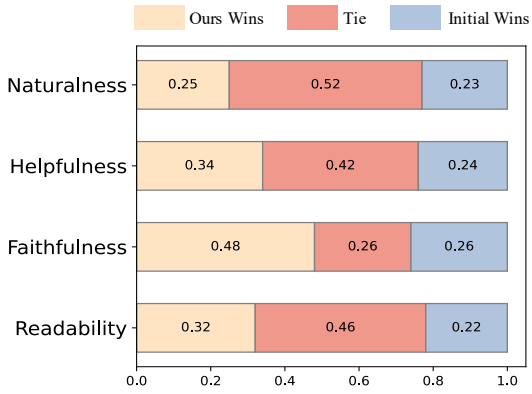

Figure 5: Human evaluation across four key dimensions.

C.4 Test-time Prompts

For baselines, the prompts for different tasks can be found in Figure 15, Figure 16, Figure 17, Figure 18, and Figure 19. To evaluate the factuality of LLMs, we modify the original FaithEval and make it a closed-book QA setting, and use the prompts shown in Figure 20. During the evaluation for CANOE, we apply the same system prompt during the Dual-GRPO training, and extract the content between <short_answer> and </short_answer> tags as the final answers for short-form generation tasks. Also, for long-form generation tasks, we extract the content between <long_answer> and </long_answer> tags as the final answers. We also find that the long-form responses generated by CANOE can provide correct answers in short-form generation tasks in the Appendix F.1. Thus, for real-world applications, we recommend using the generated long-form responses as the system responses for the user's instructions, because these long-form responses can not only faithfully complete long-form generation tasks, but also provide correct answers in short-form generation tasks.

C.5 More Detailed Experimental Results

FollowRAG contains four different QA datasets in RAG scenarios. We report the average results in Table 1. We show the more detailed results of FollowRAG in Table 7.

D Implementations Details

We implement our method based on the RL framework open-r1 (Face, 2025). We use AdamW optimizer (Loshchilov and Hutter, 2019) to train our

Model	HotpotQA		NaturalQA		TriviaQA		WebQSP	
	EM	Acc	EM	Acc	EM	Acc	EM	Acc
The state-of-the-art LLMs								
GPT-4o	24.7	32.0	37.0	55.0	62.3	72.3	44.9	71.7
GPT-4o mini	18.0	26.2	35.0	48.2	59.5	65.5	41.4	65.3
DeepSeek V3	18.7	27.7	34.9	54.3	60.0	70.0	37.1	68.9
Claude 3.7 Sonnet	15.3	24.1	33.6	53.9	62.5	72.5	33.7	64.3
OpenAI o1	27.0	34.0	37.0	50.0	63.0	76.0	35.0	68.0
DeepSeek R1	26.0	29.3	38.7	52.9	68.0	73.0	38.9	71.3
Claude 3.7 Sonnet-Thinking	20.1	30.2	35.6	53.0	63.4	72.0	36.0	66.0
LLaMA-3-Instruct Series								
LLaMA-3-Instruct-8B	13.0	18.2	31.0	40.3	45.5	60.2	35.0	60.4
LLaMA-3-Instruct-70B	24.1	28.7	36.5	45.3	63.0	66.6	31.3	42.1
SFT-8B	3.7	5.4	15.9	18.7	26.6	26.3	30.4	33.6
Context-DPO-8B	10.1	16.7	23.4	37.8	53.3	62.3	32.8	58.3
SCOPE$_{sum}$-8B	12.0	20.5	25.7	42.5	46.4	58.6	36.1	63.2
CANOE-LLaMA-8B	**21.4**	**23.3**	**37.4**	**46.9**	**60.0**	**67.3**	**44.9**	**69.3**
Qwen-2.5-Instruct Series								
Qwen-2.5-Instruct-7B	14.0	17.6	32.2	42.3	50.3	62.3	33.9	58.8
Qwen-2.5-Instruct-14B	17.5	21.7	29.3	48.0	55.6	69.3	36.9	65.7
Qwen-2.5-Instruct-32B	16.5	24.6	26.3	50.2	50.0	70.7	42.7	66.7
Qwen-2.5-Instruct-72B	21.8	28.0	34.5	51.0	61.8	73.0	35.7	70.6
SFT-7B	16.2	18.3	26.5	30.2	43.2	58.2	30.2	60.2
Context-DPO-7B	13.0	17.2	25.2	40.2	50.1	63.2	35.7	54.3
SCOPE$_{sum}$-7B	12.5	19.5	27.2	43.5	48.4	60.1	34.2	60.7
CANOE-Qwen-7B	**18.0**	**22.6**	**35.7**	**47.4**	**57.4**	**65.7**	**36.9**	**65.0**
CANOE-Qwen-14B	**19.9**	**25.7**	**41.9**	**51.6**	**63.3**	**71.7**	**59.4**	**69.3**

Table 7: Experimental results (%) on FollowRAG. **Bold** numbers indicate the best performance of models with the same model size.

model, with a 1×10^{-6} learning rate, a batch size of 14 for 7B/8B models, and a batch size of 7 for the 14B model, steering the training across two epochs. We set the maximum input length for the models to 1,024 and the maximum generation length to 1,024. The number of generations G during the RL training is set to 7, which is used in Eq. (1). We set 0.04 for β used in Eq. (1). We set 0.2 for ϵ used for the clip shown in Eq. (2). We set 0.9 for temperature in RL training to generate responses. We conduct our experiments on NVIDIA A800-80G GPUs with DeepSpeed+ZeRO2 for 7B/8B models, DeepSpeed+ZeRO2+Offloading for the 14B model, and BF16. During the inference, we set 0.7 for temperature for the evaluation of our models and baselines. For each task, we infer the model twice and report the average scores as final results.

E Human Evaluation

We conduct a human evaluation on the 90 samples from long-form generation tasks, including 30/30/30 for summarization/simplification/long-form QA. We evaluate these samples across four key dimensions: readability, faithfulness, helpfulness, and naturalness. For each comparison, three options are given (Ours Wins, Tie, and Initial Model Wins), and the majority voting determines the final result. The participants follow the principles in Figure 21 to make the decision. We invite three Ph.D. students to compare the responses generated by the models. Before participants begin to make judgments, we describe the principles of our design in detail and ensure that each participant correctly understands the principles. If the final result can not be determined by majority voting, we will hold a discussion among the participants and vote on the result again. We compare two models, including CANOE-LLaMA-8B as our method and LLaMA-3-8B as the initial model. Shown in Figure 5, we can find that our method reduces faithfulness hallucinations and also ensures the response quality for three long-form generation tasks.

F Discussion

F.1 Can Long-form Responses Generated by CANOE Provide Correct Answers in Short-form Generation Tasks?

This exploration is important because, in real-world applications, it is difficult to pre-determine whether to use generated short-form responses (i.e., the context between <short_answer> and </short_answer> tags) or long-form responses (i.e., the context between <long_answer> and </long_answer> tags) as answers to respond to user instructions. This

Model	ConFiQA Acc	FiQA Acc	CNQ Acc	FaithEval Acc	HotpotQA Acc	NaturalQA Acc	TriviaQA Acc	WebQSP Acc	Avg
The state-of-the-art LLMs									
GPT-4o	42.7	79.6	55.9	47.5	32.0	55.0	72.3	71.7	57.1
GPT-4o mini	63.7	78.8	54.3	50.9	26.2	48.2	65.5	65.3	56.6
DeepSeek V3	58.6	76.5	67.3	51.0	27.7	54.3	70.0	68.9	59.3
Claude 3.7 Sonnet	36.0	72.2	65.0	45.6	24.1	53.9	72.5	64.3	54.2
OpenAI o1	57.9	89.7	39.1	52.0	**34.0**	50.0	**76.0**	68.0	58.3
DeepSeek R1	74.3	80.7	70.2	60.1	29.3	52.9	73.0	71.3	64.0
Claude 3.7 Sonnet-Thinking	38.7	76.7	67.0	57.0	30.2	53.0	72.0	66.0	57.6
LLaMA-3-Instruct Series									
LLaMA-3-Instruct-8B	58.2	59.3	45.2	52.0	18.2	40.3	60.2	60.4	49.2
LLaMA-3-Instruct-70B	54.5	66.8	65.0	50.9	28.7	45.3	66.6	42.1	52.5
SFT-8B	70.3	59.9	65.7	43.0	5.4	18.7	26.3	33.6	40.4
Context-DPO-8B	72.9	59.5	62.3	37.5	16.7	37.8	62.3	58.3	50.9
SCOPE$_{sum}$-8B	64.6	68.7	60.6	55.7	20.5	42.5	58.6	63.2	54.3
CANOE-LLaMA-8B	80.9	84.9	73.4	74.6	23.3	46.9	67.3	69.3	65.1
- Using Generated Long-form Responses.	**92.3**	**95.5**	81.6	**78.2**	32.7	**59.3**	74.1	**79.1**	**74.1**
Δ Compared to Using Generated Short-from Response.	+11.4	+10.6	+8.2	+3.6	+9.4	+12.4	+6.8	+9.8	+9.0
Qwen-2.5-Instruct Series									
Qwen-2.5-Instruct-7B	61.0	68.4	68.2	56.1	17.6	42.3	62.3	58.8	54.3
Qwen-2.5-Instruct-14B	47.3	61.4	64.3	51.6	21.7	48.0	69.3	65.7	53.7
Qwen-2.5-Instruct-32B	66.4	81.1	66.4	47.0	24.6	50.2	70.7	66.7	59.1
Qwen-2.5-Instruct-72B	52.3	67.3	62.2	45.2	28.0	51.0	73.0	70.6	56.2
SFT-7B	69.8	76.6	65.3	50.3	18.3	30.2	58.2	60.2	53.6
Context-DPO-7B	70.6	78.2	70.1	45.7	17.2	40.2	63.2	54.3	54.9
SCOPE$_{sum}$-7B	47.9	60.9	55.3	52.3	19.5	43.5	60.1	60.7	50.0
CANOE-Qwen-7B	75.2	83.5	76.4	70.5	22.6	47.4	65.7	65.0	63.3
- Using Generated Long-form Responses.	82.9	92.3	83.2	73.2	29.8	56.9	70.6	72.7	70.2
Δ Compared to Using Generated Short-from Response.	+7.7	+8.8	+6.8	+2.7	+7.2	+9.5	+4.9	+7.7	+6.9
CANOE-Qwen-14B	87.4	88.5	84.2	67.4	25.7	51.6	71.7	69.3	68.2
- Using Generated Long-form Responses.	89.8	94.4	**87.1**	70.6	30.0	58.0	73.1	76.6	72.5
Δ Compared to Using Generated Short-from Response.	+2.4	+5.9	+2.9	+3.2	+4.3	+6.4	+1.4	+7.3	+4.2

Table 8: Experimental accuracy score results (%) on short-form generation tasks. **Bold** numbers indicate the best performance among all the models.

contrasts with the evaluation of LLMs on different datasets, as described in the test-time strategies outlined in C.4. Therefore, we first explore whether the long-form responses generated by CANOE (i.e., the context between <long_answer> and </long_answer> tags) can provide correct answers in short-form generation tasks. As shown in Table 8, when evaluating the generated long-form responses that contain the free-form answers, the accuracy scores consistently increase in all the short-form generation tasks compared to using the generated short-form responses. It also indicates that the generated short-form responses maintain conciseness, which is important for measuring the EM score, but can slightly reduce the accuracy score. Therefore, in real-world applications, we can directly use the generated long-form responses as the system responses for the user's instructions, because these long-form responses can not only efficiently and faithfully complete long-form generation tasks, but also provide correct answers in short-form generation tasks.

F.2 Final Rewards in the RL Training Stage

We show the final rewards in Table 9. We can find that models can easily learn the designed format, while accuracy and proxy rewards still remain challenging. Meanwhile, in the early stages of RL training, the format reward increases quickly and converges rapidly, and as training proceeds, the accuracy reward and the proxy reward gradually increase and eventually converge. This indicates that our well-designed training data construction strategy is effective and ensures the complexity and diversity, avoiding overfitting and reward hacking.

Model	Accuracy	Proxy	Format
CANOE-LLaMA-8B	70.3	66.1	99.4
CANOE-Qwen-7B	64.1	63.4	99.9
CANOE-Qwen-14B	83.5	76.5	100.0

Table 9: Final rewards (%) in the RL training stage.

Model	MultiFieldQA-zh	DuReader	VCSUM
LLaMA-3-Instruct-8B	80.1	65.2	42.2
CANOE-LLaMA-8B	**88.2**	**75.3**	**65.2**
Qwen-2.5-Instruct-7B	82.3	70.3	45.5
Qwen-2.5-Instruct-14B	83.5	72.2	47.8
Qwen-2.5-Instruct-32B	85.1	77.2	52.7
Qwen-2.5-Instruct-72B	88.9	80.1	57.1
CANOE-Qwen-7B	**90.1**	**78.3**	**66.5**
CANOE-Qwen-14B	**93.2**	**84.3**	**70.4**

Table 10: Results (%) on three Chinese datasets. **Bold** numbers indicate the best performance of models with the same model size.

F.3 Multilingual Transfer Ability and Context Length Generalization of CANOE

To further explore the multilingual transfer ability of CANOE, we further evaluate our model on the Chinese dataset. Specifically, we use the single-

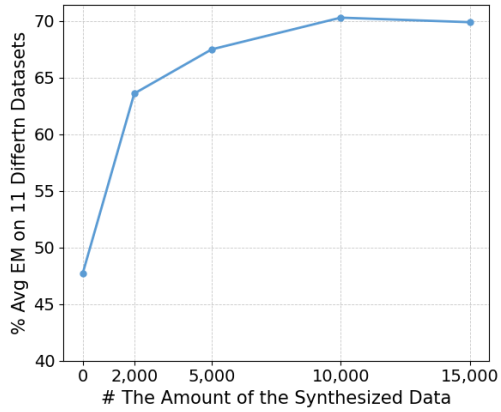

Figure 6: The Avg EM results (%) on 11 datasets with different numbers of synthesized short-form training data. We conduct the experiments based on LLaMA-3-Instruct-8B models.

document QA dataset MultiFieldQA-zh (Bai et al., 2023), the multi-document QA dataset DuReader (He et al., 2018), and the summarization dataset VCSUM (Wu et al., 2023) within LongBench (Bai et al., 2023). Following Si et al. (2024) that utilizes the GPT-4 to evaluate the correctness of QA tasks and the faithfulness of the summarization task, we use the advanced LLM to evaluate these datasets. We use the same prompts for three tasks as Si et al. (2024) to query GPT-4o. The test-time prompts for these tasks can be found in Figure 23, Figure 24, and Figure 25. As shown in Table 10, we can find that our proposed CANOE also improves the faithfulness in Chinese datasets, indicating that our proposed method has a strong multilingual transfer ability. Meanwhile, these results also indicate that our method achieves better faithfulness even when our training data is short, i.e., CANOE guarantees a consistently strong performance of the LLM in long-context scenarios.

F.4 Effect of the Amount of the Synthesized Short-form Data

To further explore the effect of the amount of the synthesized short-form data, we conduct the corresponding experiments as shown in Figure 6. We can find that increasing the number of training data improves the performance of our method, but when the number of data is greater than 10,000, the performance will be stable. For the models with different sizes, the optimal amount of training data may be different, e.g., the larger models may need more training data to achieve optimal performance. Fortunately, due to our training data construction strategy, we can simply scale and synthesize training data without human annotation.

Long-form QA Inputs:{} Question: What is the type and definition of this dance?
GRPO-LLaMA-8B: Ballet (✗)
CANOE-LLaMA-8B: Based on the passage, ballet is a type of dance that developed from lavish court spectacles in Italy and France. It is a combination of music, drama, poetry, song, costumes, and dance, and involves a combination of different elements, including music, costumes, and performance. The passage also mentions that members of the court nobility took part as performers, and that professional dancers began to take the place of court amateurs. This suggests that ballet is a type of dance that is characterized by its combination of different art forms and its performance by trained dancers. (✓)
Simplification Inputs: The Pokémon series began with the release of Pokémon Red and Green for the Game Boy in Japan. When these games proved extremely popular, an enhanced blue version was released sometime after, and the blue version was reprogrammed as Pokémon Red and Blue for international release.
GRPO-LLaMA-8B: Pocket Monsters Red and Green (✗)
CANOE-LLaMA-8B: The Pokémon series started with the release of Pokémon Red and Green in Japan, which were later enhanced and reprogrammed for international release. (✓)
Summarization Inputs: {}
GRPO-LLaMA-8B: Comedy clubs (✗)
CANOE-LLaMA-8B: Kareem Badr and his friends were able to turn around the struggling comedy club, the Hideout, by applying their skills in problem-solving and efficiency. They broke down the business into small chunks and figured out how to make each aspect better and more efficient. This approach helped them to improve the club's performance and eventually expand it. The passage also mentions that the growing popularity of stand-up comedy is fueled in part by younger audiences, who view humorous TV hosts as their primary source for news. Many comedy clubs now require talent, and there is a career and a living to be made by good comedians who are not superstars. The clubs also make money from selling drinks and dinners, as well as from cover charges and talent fees. In addition, digital media has given live comedy a boost, with one terrific joke being able to get a million hits online. (✓)

Table 11: Case study between the models trained via GRPO and our model for long-form generation tasks. Models are implemented on LLaMA-3-Instruct-8B.

F.5 Case Study between GRPO and the proposed Dual-GRPO

We find that directly applying GRPO instead of our proposed Dual-GRPO to synthesized short-form data leads to over-optimizing short-form generation and a false response generation pattern. The used system prompt for applying GRPO can be found in Figure 22. Shown in Table 11, we can find that the tuned model GRPO-LLaMA-8B tends to directly copy text spans from the given context as the final answer instead of following instructions in long-form generation tasks. However, when we apply Dual-GPRO to our synthesized data, we find that trained models can generate fluent and complete sentences. Thus, Dual-GRPO not only improves the faithfulness of LLMs in two types of response generation but also ensures the utility of models.

Relation	Description
P6	head of government
P17	country
P26	spouse
P27	country of citizenship
P30	continent
P35	head of state
P36	capital
P37	official language
P38	currency
P39	position held
P50	author
P54	member of sports team
P57	director
P86	composer
P101	field of work
P103	native language
P108	employer
P112	founder
P127	owned by
P136	genre
P1376	capital of
P140	religion
P155	follows
P159	headquarters location
P166	award received
P170	creator
P172	ethnic group
P175	performer
P178	developer
P264	record label
P276	location
P286	head coach
P407	language of work or name
P413	position played
P463	member of
P488	chairperson
P495	country of origin
P641	sport
P800	notable work
P937	work location
P169	chief executive officer

Table 12: Manually selected relations that are used to construct training data. We utilize the same manually selected relations as Bi et al. (2024).

Prompt for question generation for the samples with straightforward context.

[Instructions]
You are a sophisticated question generator. Given a triple $\{(h, r, t)\}$ collected from Wikidata, generate a question that asks about the final tail entity $\{t\}$ using the head entity $\{h\}$ and the relation $\{r\}$.

Directly give me the generated question:

Figure 7: Prompt for question generation for the samples with straightforward context.

Prompt for context generation for the samples with straightforward context.

[Instructions]
You are a sophisticated context generator. Given a triple $\{(h, r, t)\}$ collected from Wikidata, generate a brief description of the head entity $\{h\}$, approximately 150 words long. Ensure the tail entity $\{t\}$ and relation $\{r\}$ are accurately mentioned in the generated description.

Directly give me the generated context:

Figure 8: Prompt for context generation for the samples with straightforward context.

Prompt for question generation for the samples with reasoning-required context.

[Instructions]
You are a sophisticated question generator. Given a chain of triples $\{[...]\}$ collected from Wikidata, generate a question that asks about the final tail entity $\{t\}$ using the head entity $\{h\}$ and the relation $\{r\}$. Do not include any bridge entities in the question; instead, phrase the question as if directly asking about the relationship from the head entity to the tail entity

Directly give me the generated question:

Figure 9: Prompt for question generation for the samples with reasoning-required context.

Prompt for context generation for the samples with reasoning-required context.

[Instructions]
You are a sophisticated context generator. Given a chain of triples $\{[...]\}$ collected from Wikidata, generate a brief description of the head entity $\{h\}$, approximately $\{150$*n$\}$ words long. Ensure the tail entity $\{t\}$ and relation $\{r\}$ are accurately mentioned in the generated description.

Directly give me the generated context:

Figure 10: Prompt for context generation for the samples with reasoning-required context.

System prompt for Dual-GRPO.

A conversation between User and Assistant. The user gives an instruction that consists of two parts: a passage and the actual instruction, separated by two newline characters.

The passage is provided within <context> and </context> tags. The Assistant needs to refer to the given passage and complete the instruction.

The Assistant solves the question by first thinking about the reasoning process internally, according to the given passage, and then providing the response.

The response must be structured and include the following three sections, clearly marked by the respective tags:

- Reasoning Process: Explain your thought process or logical steps to derive the answer. Enclose this within <think> and </think> tags.
- Long Answer: Provide a long response that consists of syntactically and semantically complete sentences to answer the question. Enclose this within <long_answer> and </long_answer> tags.
- Short Answer: Present a concise response that directly answers the question. Enclose this within <short_answer> and </short_answer> tags.

Format your response exactly as follows:
<think> reasoning process here. </think> <long_answer> detailed answer here. </long_answer> <short_answer> the concise answer here. </short_answer>.

Figure 11: System prompt for Dual-GRPO.

Prompt used to calculate quality score for text summarization.

You are asked to evaluate the quality of the AI assistant's generated summary as an impartial judge, and your evaluation should take into account factors including readability (whether the summary is clear and easy to understand) and coherence (whether the assistant's summary is logical and orderly).

Read the AI assistant's summary and input passages, and give an overall integer rating in on a scale of 1 to 5, where 1 is the lowest and 5 is the highest based on the evaluation criteria, strictly in the following format:"[[rating]]", e.g. "[[5]]".

Input Passages: {}
Assistant's summary:{}
Rating:

Figure 12: Prompt used to calculate quality score for text summarization.

Prompt used to calculate quality score for text simplification.

You are asked to evaluate the quality of the AI assistant's generated text simplification as an impartial judge, and your evaluation should take into account factors including readability (whether the simplification is clear and easy to understand) and coherence (whether the assistant's simplification is logical and orderly).

Read the AI assistant's simplified version and the original text, and give an overall integer rating on a scale of 1 to 5, where 1 is the lowest and 5 is the highest based on the evaluation criteria, strictly in the following format: "[[rating]]", e.g. "[[5]]".

Original text: {}
AI assistant's simplification: {}
Rating:

Figure 13: Prompt used to calculate quality score for text simplification.

Prompt used to calculate quality score for long-form QA.

You are asked to evaluate the quality of the AI assistant's generated long-form answer as an impartial judge, and your evaluation should take into account factors including readability (whether the answer is clear and easy to understand) and coherence (whether the answer is logical and well-organized).

Read the AI assistant's long-form answer and the original question, and give an overall integer rating on a scale of 1 to 5, where 1 is the lowest and 5 is the highest, based on the evaluation criteria, strictly in the following format: "[[rating]]", e.g., "[[5]]".

Question: {}
Assistant's long-form answer: {}
Rating:

Figure 14: Prompt used to calculate quality score for long-form QA.

Test-time prompt used for short-form QA tasks.

Passages: {}

Refer to the passages above and answer the following question with just a few words.

Question: {}

Answer:

Figure 15: Test-time prompt used for short-form QA tasks.

Test-time prompt used for multiple-choice QA task.

Passages: {}

Refer to the passages above and answer the following question with just a few words.

Question: {}

Please select the correct option according to the question, and output the option letter (e.g. A/B/C/D):

Options: {}

Answer:

Figure 16: Test-time prompt used for multiple-choice QA task.

Test-time prompt used for text summarization.

Passage: {}

Refer to the passage above and provide a summary as the response.

Summary:

Figure 17: Test-time prompt used for text summarization.

Test-time prompt used for text simplification.

Passage: {}

Refer to the passage above and simplify it to improve its readability, ensuring its core meaning remains intact. Please provide only the simplified text as the response.

Simplified text:

Figure 18: Test-time prompt used for text simplification.

Test-time prompt used for long-form QA task.

Passage: {}

Refer to the passages above and answer the following question.

Question: { }

Figure 19: Test-time prompt used for long-form QA task.

Test-time prompt used for FaithEval in closed-book QA settings.

Question: {}

Please select the correct option according to the question, and output the option letter (e.g. A/B/C/D):

Options: {}

Answer:

Figure 20: Test-time prompt used for FaithEval in closed-book QA settings.

The principles of human evaluation for long-form responses generation.

You are asked to evaluate the responses generated by different models. You should choose the preferred responses according to the following perspectives independently:

1. **Readability**: Whether the response is clear and easy to understand?

2. **Faithfulness**: Whether the response is faithful to the context and the information can be grounded in the provided context.

3. **Helpfulness**: Whether the response provides useful information and follows the instructions from users?

4. **Naturalness**: Whether the response sounds natural and fluent?

Finally, please make a decision among the 3 opinions, including Win, Tie, and Loss.

Figure 21: The principles of human evaluation for long-form responses generation.

System prompt for GRPO in the ablation study.

A conversation between User and Assistant. The user gives an instruction that consists of two parts: a passage and the actual instruction, separated by two newline characters.

The passage is provided within <context> and </context> tags. The Assistant needs to refer to the given passage and complete the instruction.

The Assistant solves the question by first thinking about the reasoning process internally, according to the given passage, and then providing the response.

The response must be structured and include the following two sections, clearly marked by the respective tags:

- Reasoning Process: Explain your thought process or logical steps to derive the answer. Enclose this within <think> and </think> tags.
- Answer: Present a concise response that directly answers the question. Enclose this within <answer> and </answer> tags.

Format your response exactly as follows:
<think> reasoning process here. </think> <answer> answer here. </answer>.

Figure 22: System prompt for GRPO in the ablation study.

Test-time prompt used for MultiField-zh.

阅读以下文字并用中文简短回答：{}
现在请基于上面的文章回答下面的问题，只告诉我答案，不要输出任何其他字词。
问题：{}
回答：

Figure 23: Test-time prompt used for MultiField-zh.

Test-time prompt used for DuReader.

请基于给定的文章回答下述问题。
文章：{}
问题：{}
回答：

Figure 24: Test-time prompt used for DuReader.

Test-time prompt used for VCSUM.

下面有一段会议记录，请你阅读后，写一段总结，总结会议的内容。
会议记录：{}
会议总结：

Figure 25: Test-time prompt used for VCSUM.