CORE: INTERNALIZING CONTEXT FAITHFULNESS VIA INTERVENTION-DERIVED PROCESS REWARDS IN REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In context-grounded tasks, Large Language Models may underutilize provided contextual information and instead rely more heavily on their parametric knowledge. Although reinforcement learning can optimize models toward task-specific objectives, outcome-based rewards provide only sparse supervision and offer limited direct guidance on how the context should be used during generation. Process-based rewards provide finer-grained supervision over intermediate generation steps, but often rely on costly intermediate annotations. We propose CORE, a reinforcement learning framework for improving contextual faithfulness during generation. Given a trajectory generated with access to the context, CORE re-evaluates the likelihood of the same generated tokens after intervening on context availability and uses the resulting token-level likelihood changes to measure contextual influence. This signal is incorporated as a dense, annotation-free process reward during policy optimization. To mitigate reward exploitation through excessive context copying, CORE further introduces an accuracy-gating mechanism that applies the contextual influence reward only to correct trajectories, coupling contextual reliance with task correctness. Across multiple model architectures, reinforcement learning algorithms, and both in-distribution and out-of-distribution benchmarks, CORE consistently improves task accuracy over the corresponding baselines. Complementary LLM-as-a-Judge evaluations further indicate improved contextual faithfulness on the evaluated settings.

1 INTRODUCTION

The advent of Large Language Models (LLMs) (Grattafiori et al., 2024; Bai et al., 2023; Achiam et al., 2023; Bi et al., 2024; Team et al., 2025) has revolutionized natural language processing, particularly in tasks requiring external knowledge integration. In these context-grounded tasks, the reliability of the model critically depends on its ability to faithfully extract and reason over the provided information. However, despite their impressive capabilities, LLMs suffer from a pervasive parametric bias (Longpre et al., 2021): when the provided context contradicts the model's internal pre-trained memory, or when the required reasoning involves long-tail knowledge, the model may underutilize the explicit textual evidence and resort to its internalized priors. This tendency leads to context faithfulness failures (Bi et al., 2025; Qi et al., 2025), severely undermining the trustworthiness of LLMs in context-related applications.

Reinforcement learning (RL) methods, such as PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024), provide a general framework for optimizing LLM behavior toward task-specific objectives. However, standard RL optimization relies predominantly on sparse outcome-based rewards, which evaluate only the correctness of the final response. In context-grounded tasks, this sparse signal poses a critical challenge: it indicates whether the generated answer is correct, but fails to verify the underlying reasoning process. Consequently, a model may obtain correct answers through parametric memory or dataset priors without sufficiently relying on the provided context. Because the outcome reward does not directly penalize this misalignment, standard outcome-based RL provides limited direct supervision for contextual adherence. On the other hand, while recent inference-time interventions, such as Context-Aware Decoding (Shi et al., 2024), attempt to enforce context dependency via contrastive log-probability shifts, they incur significant computational latency at ev-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

ery generated token. This introduces additional inference-time computation at each decoding step, which can become costly for large-scale or long-form generation.

Rather than treating contextual reliance as an inference-time correction or leaving it as an indirect consequence of sparse outcome optimization, we seek to internalize this capability into the policy parameters. Recent advances in process-based reinforcement learning have shown that intermediate feedback can provide more fine-grained supervision than outcome-only rewards (Khalifa et al., 2025; Cui et al., 2025; Li & Li, 2025). However, obtaining explicit process-level annotations is costly and difficult to scale. We therefore ask whether contextual reliance can instead be characterized using the model's own token probabilities. Given a trajectory generated with access to the context, we re-evaluate the likelihood of the same generated tokens under a controlled intervention on context availability while keeping the generated prefix fixed. The resulting change in token log-probability provides a fine-grained measure of contextual influence at each generation step. By incorporating this signal as an annotation-free process reward during reinforcement learning, we directly encourage the policy to internalize contextual reliance, rather than enforcing it through additional interventions at inference time.

Building upon this insight, we introduce CORE (**CO**ntextual **RE**liance RL), a reinforcement learning framework designed to encourage contextual reliance during policy optimization. For each rollout phase, CORE performs an additional forward pass after intervening on context availability to estimate token-level contextual influence. However, directly optimizing contextual influence can encourage reward exploitation through excessive context copying, potentially increasing process scores without improving task-relevant reasoning. To mitigate this risk, CORE incorporates an accuracy-gating mechanism. Specifically, the trajectory-averaged contextual influence score is incorporated into the reward only when the final answer is correct. This composite reward formulation is agnostic to the underlying optimization algorithm and seamlessly integrates into prevalent RL paradigms, such as PPO and GRPO. Consequently, CORE encourages contextual reliance during training without introducing additional inference-time decoding operations.

We conduct extensive experiments across diverse reasoning and reading-comprehension benchmarks using multiple model architectures. Our main contributions are as follows:

- We propose CORE, an RL framework that derives a token-level contextual influence signal by measuring changes in generated-token likelihood under an intervention on context availability. This signal provides a dense, annotation-free process reward for encouraging contextual reliance during policy optimization.
- We introduce an accuracy-gating mechanism that applies the contextual influence reward only to correct trajectories, coupling contextual reliance with task correctness and empirically mitigating excessive context-copying behavior.
- Across multiple model architectures and both in-distribution and out-of-distribution benchmarks, CORE consistently improves task accuracy over the corresponding RL baselines. Complementary LLM-as-a-Judge (Gu et al., 2024) evaluations further indicate improved contextual faithfulness on the evaluated settings.

2 Related Work

2.1 Context awareness in LLMs

Large language models often exhibit a strong prior toward information encoded during pre-training. Longpre et al. (2021) categorized entity-based knowledge conflicts, demonstrating that when a provided context contradicts a model's internalized facts, the model may favor its parametric knowledge over the supplied context. Subsequent research (Si et al., 2026; Xu et al., 2025) further showed that this tendency reflects persistent challenges in contextual adherence, where models may struggle to prioritize input-side constraints over their parametric priors.

To address this issue, researchers have explored methods for strengthening the contribution of contextual information during generation. Representative inference-time approaches such as CAD (Shi et al., 2024) and DAGCD (Huang et al., 2025) steer generation toward context-consistent outputs using contrastive likelihood signals or internal uncertainty estimates. Because these methods operate during decoding, they require additional inference-time computation at generation time. Be-

yond inference-time steering, Context-DPO (Bi et al., 2025) addresses context faithfulness through preference-based alignment, encouraging models to favor responses that are consistent with the provided context. This demonstrates that contextual adherence can also be improved through training-time alignment rather than decoding-time intervention. In contrast to preference-level supervision, CORE targets contextual reliance at a finer process level. It measures token-level contextual influence by intervening on context availability and uses the resulting likelihood changes as an annotation-free process reward during reinforcement learning. In this way, CORE directly encourages contextual reliance to be internalized into the policy without requiring auxiliary decoding operations at inference time.

2.2 Process-based Reinforcement Learning

Reinforcement learning has become a widely used paradigm for optimizing LLM behavior (Ouyang et al., 2022; Ahmadian et al., 2024; Yu et al., 2026), with algorithms commonly optimizing rewards derived from final response quality. However, outcome-based rewards provide sparse supervision and can make credit assignment difficult, particularly when a correct final answer does not reveal whether the model actually relied on the provided context (Xu et al., 2024).

To provide more fine-grained guidance, process-based rewards have been introduced to supervise intermediate reasoning steps (Lightman et al., 2023). More recently, implicit and endogenous rewards have emerged as scalable alternatives to externally annotated process supervision. EndoRM (Li et al., 2025) develops a theoretical connection between next-token prediction and implicit reward modeling, interpreting model logits through a soft Q-function perspective. PRIME (Cui et al., 2025) proposes an RL framework that derives implicit process rewards from policy rollouts, avoiding the need to train a separate reward model. Unlike PRIME, which derives implicit process rewards for general reasoning quality, CORE focuses specifically on contextual reliance. For each generated trajectory, CORE intervenes on context availability and measures the resulting change in the likelihood of the generated tokens. This token-level contextual influence signal is then used as an annotation-free process reward, providing direct training feedback on how strongly generation depends on the supplied context.

3 Methodology

3.1 Preliminaries

Notation and Task Formalization. We consider a context-grounded generation task where the model is provided with a context c and an instruction x. Let π_θ denote a language model parameterized by θ. The model generates a response $y = (y_1, y_2, \ldots, y_T)$ token by token, according to the conditional probability $\pi_\theta(y|c,x) = \prod_{t=1}^{T} \pi_\theta(y_t|c,x,y_{<t})$. In RL-based post-training of LLMs, the policy is typically optimized to maximize expected reward, often with a KL regularization term that discourages excessive deviation from a reference policy π_{ref}.

PPO. Proximal Policy Optimization (Schulman et al., 2017) optimizes the policy by maximizing a clipped surrogate objective. Let $w_t(\theta) = \frac{\pi_\theta(y_t|c,x,y_{<t})}{\pi_{\theta_{\text{old}}}(y_t|c,x,y_{<t})}$ be the probability ratio. The objective is:

$$\mathcal{J}_{\text{PPO}}(\theta) = \mathbb{E}[\min(w_t(\theta)\hat{A}_t, \text{clip}(w_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)], \tag{1}$$

where ϵ is a hyperparameter for clipping and $\hat{A}_t$ is the advantage estimated by a critic model V_ϕ. The total reward R (including KL penalty) is used to compute $\hat{A}_t$.

GRPO. Group Relative Policy Optimization (Shao et al., 2024) optimizes the policy by sampling a group of G responses. Let $w_t(\theta) = \frac{\pi_\theta(y_t^{(i)}|c,x,y_{<t}^{(i)})}{\pi_{\theta_{\text{old}}}(y_t^{(i)}|c,x,y_{<t}^{(i)})}$ be the probability ratio. The objective function is formulated as:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\Big(\min(w_t(\theta)\hat{A}_i, \text{clip}(w_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_i)\Big) - \beta\mathbb{D}_{\text{KL}}(\pi_\theta\|\pi_{\text{ref}}), \tag{2}$$

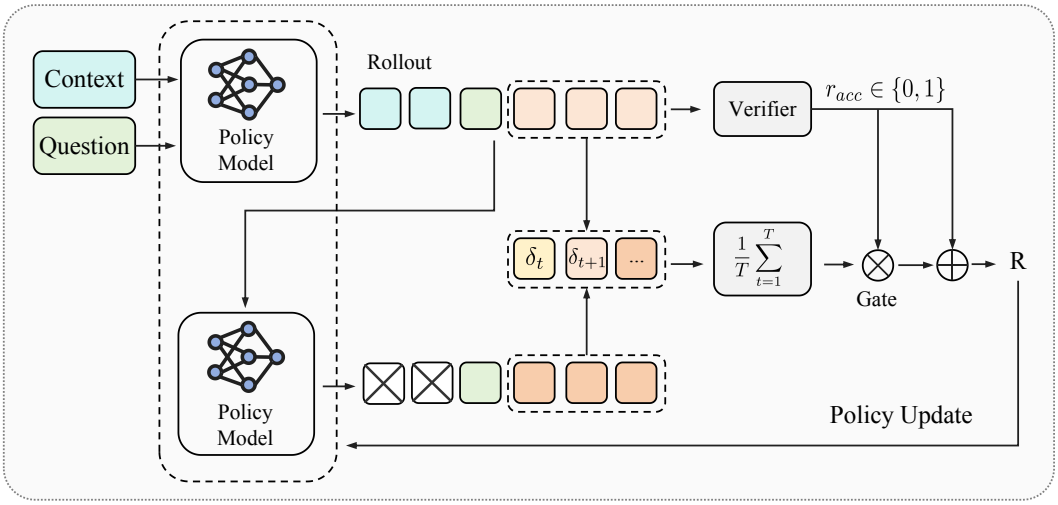


Figure 1: Overview of the proposed CORE framework. Given a context and a question, the policy model generates a rollout trajectory. The same generated trajectory is then re-evaluated under a context-removed condition to compute token-level contextual influence scores δ_t from changes in generated-token likelihood. These scores are aggregated over the trajectory, gated by task correctness, and combined with the accuracy reward for policy optimization, thereby encouraging contextual reliance to be internalized into the policy during training.

where ϵ is the clipping hyperparameter and β is the KL penalty coefficient. The advantage $\hat{A}_i$ is computed via group-relative reward normalization:

$$\hat{A}_i = \frac{R_i - \text{mean}\left(\{R_j\}_{j=1}^{G}\right)}{\text{std}\left(\{R_j\}_{j=1}^{G}\right) + \varepsilon_{\text{norm}}}, \tag{3}$$

where $R_i = R(c, x, y^{(i)})$ denotes the reward assigned to the i-th response, and $\varepsilon_{\text{norm}} > 0$ is a small constant for numerical stability.

3.2 CORE

To enhance the model's faithfulness to the provided context, we propose CORE, a reinforcement learning framework that optimizes the model's adherence to input contexts. The core of our approach is to provide a fine-grained process-level reward that isolates and reinforces the influence of the context c on the generated response.

Quantifying Token-Level Contextual Influence. A key challenge in context-grounded generation is to determine how strongly the generated response depends on the provided context c. To address this, we define a token-level contextual influence score that measures the sensitivity of each generated token to context availability, i.e., how much the likelihood of the observed token changes when the provided context is removed while the generated prefix is kept fixed.

During the RL rollout phase, given a context c and an instruction x, the current policy π_θ first generates a trajectory y. We then perform a counterfactual intervention by masking the context c. We compute the token-level influence δ_t as the difference between the log-probability of y_t conditioned on the full input and its log-probability under the context-removed condition:

$$\delta_t = \log \pi_\theta(y_t | c, x, y_{<t}) - \log \pi_\theta(y_t | x, y_{<t}). \tag{4}$$

A larger δ_t indicates that the observed token is more strongly supported by the context relative to the context-removed condition. By evaluating δ_t along the generated trajectory, we obtain a dense signal that characterizes how strongly individual generation steps depend on the provided context.

Accuracy-Gated Reward Formulation. While contextual influence provides a direct signal of context reliance, optimizing this signal alone may encourage reward exploitation through excessive context copying or superficial lexical overlap, without necessarily improving task correctness. To

balance contextual reliance with task performance, we introduce an accuracy-gating mechanism that activates the contextual influence reward only for correct trajectories.

We first define the task-specific accuracy reward as

$$r_{\text{acc}} = \begin{cases} 1, & \text{if } y \text{ is correct} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

We then aggregate the token-level contextual influence scores over the generated trajectory and gate the resulting score by r_{acc}:

$$r_{\text{CORE}}(c, x, y) = r_{\text{acc}} \cdot \left(\frac{1}{T} \sum_{t=1}^{T} \delta_t \right). \tag{6}$$

This formulation ensures that the model is only reinforced for context reliance when it leads to a correct answer. If the response is incorrect, the reward is nullified, thereby preventing the model from over-relying on the context to justify erroneous reasoning.

Optimization. The final reward R optimized during the reinforcement learning process is a combination of the task-specific accuracy reward r_{acc} and the CORE reward r_{CORE}. We define the composite reward as follows:

$$R(c, x, y) = r_{\text{acc}} + \alpha \cdot r_{\text{CORE}}, \tag{7}$$

where α is a hyperparameter that controls the strength of the contextual reliance incentive.

The resulting reward can be directly incorporated into standard policy-optimization algorithms. For PPO, $R(c, x, y)$ is used to estimate the advantage $\hat{A}_t$, whereas for GRPO, it is used to compute the group-relative advantage through reward normalization. Together, r_{acc} provides outcome-level supervision for task correctness, while r_{CORE} provides process-level supervision for contextual reliance. This allows CORE to encourage contextual reliance to be internalized during training without introducing auxiliary decoding operations at inference time.

4 Experiments

In this section, we present a comprehensive empirical evaluation of the proposed CORE framework. We detail the experimental setup, including the datasets, base models, and training hyperparameters in Section 4.1, main accuracy results (Section 4.2), LLM-as-Judge faithfulness evaluation (Section 4.3), ablation studies (Section 4.4), and training dynamics analysis (Section 4.5).

4.1 Setup

Benchmarks. To comprehensively evaluate the effectiveness of our proposed context-reliance reward, we design an experimental setting that assesses both Independent and Identically Distributed (IID) and Out-of-Distribution (OOD) performance. All selected benchmarks involve reasoning or answering with access to provided contextual information, making them suitable for evaluating contextual reliance. For the training phase, we construct a consolidated dataset consisting of 9,000 samples. This is achieved by randomly sampling 3,000 samples from the respective training splits of LogiQA (Liu et al., 2020), Cladder (Jin et al., 2023), and ReClor (Yu et al., 2020). For the evaluation phase, our IID test sets consist of the test splits from the aforementioned three datasets (LogiQA, Cladder, ReClor). To rigorously test the models' zero-shot generalization and context faithfulness on unseen tasks, we incorporate four diverse OOD datasets: BoolQ (Clark et al., 2019), SQuAD (Rajpurkar et al., 2016), MemoTrap (Liu & Liu, 2023), and IfQA (Yu et al., 2023).

Models. To verify the generalizability of CORE, we conduct experiments across various model families and parameter scales. Specifically, we utilize the Qwen3 series (1.7B, 4B, and 8B) to assess scalability, and further evaluate our approach on Llama-3.1-8B-Instruct to demonstrate its broad applicability across different architectures.

Implementation Details. Since our proposed method functions as a process-level reward mechanism, it is fundamentally agnostic to the underlying optimization algorithm. Therefore, we implement and validate our method on two of the most prevalent reinforcement learning paradigms for Large Language Models, PPO and GRPO. All training procedures are conducted on a single node

Table 1: Accuracy results on IID and OOD benchmarks. We report the exact accuracy(%) across various model scales and RL paradigms. **Bold** indicates the best performance within each specific model and algorithm block.

Model	Method	IID			OOD				Average
		LogiQA	Cladder	ReClor	BoolQ	SQuAD	MemoTrap	IfQA	
Qwen3-1.7B	Zero Shot	52.84	67.68	69.80	81.04	52.80	88.03	64.00	68.03
	CAD	48.54	68.34	66.00	**81.86**	53.06	88.78	62.71	67.04
	PPO	54.07	68.94	69.80	80.40	52.22	88.68	62.71	68.12
	PPO+Ours	**56.68** +2.61	71.69 +2.75	**72.00** +2.20	81.16 +0.76	**54.52** +2.30	91.45 +2.77	64.86 +2.15	70.34 +2.22
	GRPO	51.31	78.00	**71.00**	80.49	51.88	85.15	62.43	68.61
	GRPO+Ours	54.69 +3.38	**80.78** +2.78	71.00 +0.00	81.44 +0.95	53.99 +2.11	**92.09** +6.94	**65.57** +3.14	**71.36** +2.75
Qwen3-4B	Zero Shot	67.13	70.11	83.60	85.47	57.46	83.76	72.86	74.34
	CAD	61.14	70.64	80.00	86.36	57.12	72.86	71.43	71.36
	PPO	67.74	71.05	85.00	84.92	57.31	81.84	72.14	74.29
	PPO+Ours	69.43 +1.69	71.43 +0.38	**86.80** +1.80	86.39 +1.47	57.87 +0.56	**86.97** +5.13	**73.43** +1.29	76.04 +1.75
	GRPO	67.59	83.90	84.20	85.23	57.22	80.13	72.14	75.77
	GRPO+Ours	**70.51** +2.92	**86.96** +3.06	85.60 +1.40	**86.79** +1.56	**58.90** +1.68	86.43 +6.30	73.29 +1.15	**78.35** +2.58
Qwen3-8B	Zero Shot	70.20	70.45	89.60	85.20	57.29	81.84	73.14	75.39
	CAD	64.36	70.08	86.40	86.48	56.92	75.75	72.14	73.16
	PPO	72.50	68.18	89.60	86.48	56.78	79.06	73.57	75.17
	PPO+Ours	**73.12** +0.62	75.57 +7.39	91.40 +1.80	**87.16** +0.68	58.53 +1.75	**91.24** +12.18	**74.29** +0.72	78.76 +3.59
	GRPO	71.73	84.18	91.00	85.50	57.38	83.23	73.14	78.02
	GRPO+Ours	72.66 +0.93	**86.84** +2.66	**91.60** +0.60	86.67 +1.17	**58.80** +1.42	87.93 +4.70	**74.29** +1.15	**79.82** +1.80
Llama-3.1-8B-Instruct	Zero Shot	33.79	56.34	48.20	83.15	52.86	77.24	63.71	59.33
	CAD	45.31	59.53	61.40	83.70	54.02	80.24	63.43	63.95
	PPO	43.63	64.46	57.80	80.46	52.37	67.52	63.14	61.34
	PPO+Ours	45.78 +2.15	67.93 +3.47	62.80 +5.00	84.83 +4.37	56.04 +3.67	**88.35** +20.83	68.14 +5.00	67.70 +6.36
	GRPO	49.31	80.02	68.00	83.52	52.21	72.01	65.14	67.17
	GRPO+Ours	**53.15** +3.84	**81.28** +1.26	**71.60** +3.60	**84.89** +1.37	**56.40** +4.19	85.36 +13.35	**68.43** +3.29	**71.59** +4.42

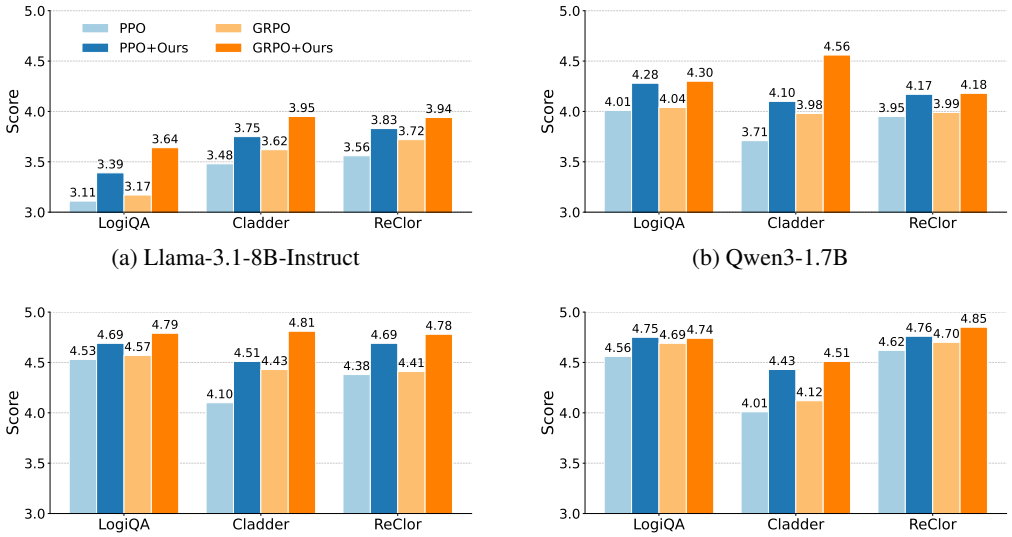
equipped with 8 NVIDIA A100 GPUs. Due to the substantial memory footprint required by the actor-critic architecture of PPO, we employ LoRA (Hu et al., 2022) with a rank of 64 for models larger than 3B parameters (i.e., Qwen3-4B, Qwen3-8B, and Llama-3.1-8B-Instruct). Conversely, for the Qwen3-1.7B model and all GRPO experiments, we perform full-parameter fine-tuning to maximize policy expressiveness. For the CAD baseline, we set the decoding coefficient to $\lambda_{\text{CAD}} = 0.5$, following the recommended setting in the original paper. For all evaluations, we adopt greedy decoding. Further comprehensive training details are deferred to Appendix A.

4.2 Main Results

We first evaluate whether optimizing for contextual reliance can directly translate into downstream performance gains. Table 1 reports the accuracy of our method applied to two mainstream RL algorithms (PPO and GRPO) across four different base models. Across all evaluated model architectures and both RL algorithms, incorporating the CORE reward improves the corresponding RL baseline in terms of average accuracy.

For example, on Qwen3-8B, CORE improves PPO from 75.17% to 78.76% (+3.59 points) and GRPO from 78.02% to 79.82% (+1.80 points). The gains are particularly pronounced on Llama-3.1-8B-Instruct, where PPO+CORE and GRPO+CORE improve the corresponding baselines by 6.36 and 4.42 points in average accuracy, respectively. These results indicate that the contextual-reliance reward can be incorporated into different policy-optimization algorithms and remains effective across the evaluated model families and scales. Compared with the inference-time CAD baseline, CAD yields mixed performance across model architectures: it improves over zero-shot decoding in some settings, but does not consistently improve average accuracy across the evaluated models. In contrast, CORE consistently improves the corresponding PPO and GRPO baselines while introducing no additional decoding operations at inference time.

On the IID datasets (LogiQA, Cladder, and ReClor), CORE generally improves the corresponding RL baselines across different models. For example, under GRPO, Qwen3-4B improves from 83.90% to 86.96% on Cladder, while Llama-3.1-8B-Instruct improves across all three IID datasets.



(a) Llama-3.1-8B-Instruct (b) Qwen3-1.7B

(c) Qwen3-4B (d) Qwen3-8B

Figure 2: Contextual faithfulness evaluation using Qwen3-235B-A22B-Instruct-2507 as the judge. We report average faithfulness scores on a scale from 1 to 5, where higher scores indicate stronger consistency with the provided context, across three IID datasets and four base models. Incorporating CORE into PPO and GRPO consistently improves the judged contextual faithfulness over the corresponding RL baselines across the evaluated settings.

These results suggest that directly encouraging contextual reliance during policy optimization can complement outcome-based supervision on context-grounded reasoning tasks.

The improvements are also observed on the OOD benchmarks, with particularly large gains on MemoTrap, which is designed to evaluate whether models follow the provided context when it conflicts with memorized knowledge. Standard PPO performs below the corresponding zero-shot model on MemoTrap in several settings; for example, Qwen3-8B decreases from 81.84% to 79.06%, while Llama-3.1-8B-Instruct decreases from 77.24% to 67.52%. Incorporating CORE substantially improves performance in these cases: PPO+CORE reaches 91.24% on Qwen3-8B (+12.18 points over PPO) and 88.35% on Llama-3.1-8B-Instruct (+20.83 points over PPO).

These substantial gains on MemoTrap, together with consistent improvements on BoolQ, SQuAD, and IfQA, provide strong evidence that the contextual reliance encouraged by CORE transfers beyond the training benchmarks. In particular, the results on MemoTrap suggest improved robustness in unseen settings where parametric knowledge may conflict with the provided context, indicating that CORE promotes transferable context-grounded behavior.

4.3 LLM-AS-JUDGE RESULTS

Beyond conventional benchmark-based evaluation metrics, we further employ an LLM-as-a-judge paradigm to provide a more intuitive assessment of faithfulness to contextual information. We use Qwen3-235B-A22B-Instruct-2507 (Yang et al., 2025) as the evaluator. The judge follows a fixed rubric designed to assess contextual faithfulness, with particular emphasis on knowledge conflicts between the response and the provided context. It penalizes responses that contradict or misrepresent contextual information, while allowing supplementary reasoning that remains consistent with the context. The judge assigns a faithfulness score from 1 (severe contradiction) to 5 (no conflict with the provided context). The complete evaluation prompt is provided in Appendix B.

As illustrated in Figure 2, incorporating CORE consistently improves the judged faithfulness scores across the evaluated models and the three IID datasets. For example, on Llama-3.1-8B-Instruct, CORE improves the scores across LogiQA, Cladder, and ReClor under both PPO and GRPO. Under GRPO, the faithfulness score on ReClor increases from 3.64 to 3.94. These results indicate that the improvements obtained with CORE are accompanied by stronger consistency with the provided

Table 2: Ablation study on the reward scaling factor α. The evaluation is conducted on the Qwen3-1.7B model across the various datasets. We report accuracy (%) under both PPO and GRPO. The average accuracy peaks at $\alpha = 0.1$ for both optimization algorithms.

	PPO				GRPO			
α	LogiQA	Cladder	ReClor	Avg.	LogiQA	Cladder	ReClor	Avg.
α=0.05	55.91	69.95	71.40	65.75	52.53	**81.22**	69.40	67.72
α=0.1	56.68	**71.69**	**72.00**	**66.79**	**54.69**	80.78	**71.00**	**68.82**
α=0.2	**57.45**	71.21	70.00	66.22	50.54	80.93	69.20	66.89
α=0.3	55.30	68.75	68.60	64.22	50.23	79.32	67.20	65.58
α=0.5	49.46	68.47	62.00	59.98	50.54	78.50	66.40	65.15

context. The comparison with standard RL further highlights that task accuracy and contextual faithfulness are not necessarily equivalent. For instance, on Qwen3-1.7B and Cladder, standard GRPO obtains a faithfulness score of 3.98, whereas GRPO+CORE increases it to 4.56. Similar improvements are observed for larger models; on Qwen3-8B and ReClor, GRPO+CORE reaches a score of 4.85. Taken together, these results suggest that outcome-only optimization provides limited direct supervision for contextual adherence, whereas explicitly rewarding contextual reliance can improve faithfulness while maintaining strong task performance.

4.4 Ablation Study

In our composite reward formulation (Eq. 7), the hyperparameter α dictates the magnitude of the contextual reliance incentive relative to the sparse task-accuracy reward. To investigate the sensitivity of our framework to this scaling factor and determine its optimal configuration, we conduct an ablation study using the Qwen3-1.7B model on LogiQA, Cladder and ReClor datasets. Table 2 presents the performance variations across both GRPO and PPO paradigms when $\alpha \in \{0.05, 0.1, 0.2, 0.3, 0.5\}$. Setting $\alpha = 0.1$ consistently achieves the best average accuracy for both GRPO (68.82) and PPO (66.79). At this value, the CORE reward effectively guides the model to utilize the context without interfering with the primary objective of answering the question correctly. Consequently, we highly recommend and adopt $\alpha = 0.1$ as the default setting for all main experiments in Section 4.2.

Table 3: Effect of accuracy gating on lexical overlap and task accuracy using Qwen3-1.7B. The 1-gram and 3-gram overlap rates are evaluated on LogiQA at the token level, while average accuracy is computed across all evaluated benchmarks. Removing the accuracy gate increases lexical overlap while substantially reducing average task accuracy.

Ablation	1-gram overlap rate(%)	3-gram overlap rate(%)	Average accuracy
GRPO	45.24	7.07	68.61
CORE w/o gate	49.21	9.83	63.78
Ours	45.43	6.41	71.36

We further examine whether the accuracy gate mitigates reward exploitation through excessive lexical copying. We compute token-level overlap on LogiQA using Qwen3-1.7B. The 1-gram overlap rate measures the proportion of response tokens whose token IDs also appear in the input, while the 3-gram overlap rate measures the proportion of contiguous three-token sequences in the response that also occur in the input. To assess the corresponding effect on task performance, we additionally report the average accuracy across all evaluated benchmarks. As shown in Table 3, removing the accuracy gate increases the 1-gram overlap rate from 45.43% to 49.21% and the 3-gram overlap rate from 6.41% to 9.83%, while reducing the average accuracy from 71.36% to 63.78%. These results suggest that, without correctness gating, directly optimizing contextual influence can encourage greater lexical reuse without corresponding improvements in task performance.

Compared with standard GRPO, CORE maintains a similar 1-gram overlap rate (45.24% vs. 45.43%) while reducing the 3-gram overlap rate from 7.07% to 6.41% and improving the average accuracy from 68.61% to 71.36%. This indicates that the performance improvements of CORE



(a) Accuracy Reward

(b) CORE Reward

Figure 3: Training dynamics of Ours and standard GRPO on Qwen3-1.7B. (a) Ours maintains a higher accuracy reward over most of the training process. (b) Ours exhibits a substantially higher and steadily increasing contextual influence score, whereas the corresponding diagnostic score of standard GRPO remains lower with little upward trend.

are not associated with increased copying of longer spans from the input. Overall, the accuracy gate helps couple contextual reliance with task correctness while mitigating excessive lexical reuse.

4.5 Training Dynamics Analysis

To better understand the optimization behavior underlying the final performance metrics, we visualize the training curves of both the accuracy reward and the CORE process reward. Crucially, while training the standard GRPO baseline, we silently calculate and log its contextual influence score as a diagnostic metric, without feeding it back to the optimizer.

Figure 3(a) shows that GRPO+CORE achieves consistently higher accuracy reward throughout training, demonstrating that the CORE reward not only preserves but actively enhances the primary optimization objective. This confirms that rewarding contextual reliance and optimizing for task accuracy are complementary rather than competing goals.

Figure 3(b) shows a clearer difference in contextual influence. The CORE score of GRPO+CORE increases steadily during training, whereas the corresponding diagnostic score of standard GRPO remains substantially lower and exhibits little upward trend. This suggests that outcome-only optimization does not directly encourage contextual reliance, while CORE explicitly provides such a training signal. Together, these dynamics are consistent with the final evaluation results, indicating that CORE increases contextual reliance during policy optimization rather than treating it as an incidental consequence of task accuracy.

5 Conclusion

In this work, we proposed CORE, a reinforcement learning framework for improving context faithfulness in context-grounded generation. CORE measures token-level contextual influence through changes in generated-token likelihood under an intervention on context availability, and incorporates this signal as a dense, annotation-free process reward. An accuracy-gating mechanism further couples contextual reliance with task correctness by applying the contextual influence reward only to correct trajectories. Across multiple model architectures and both in-distribution and out-of-distribution benchmarks, CORE consistently improves task accuracy over standard RL baselines, while complementary LLM-as-a-Judge evaluations indicate improved contextual faithfulness. The consistent improvements under both PPO and GRPO further demonstrate that the proposed process reward can be incorporated into different policy-optimization paradigms. These results suggest that directly optimizing contextual reliance at the process level can help internalize context-faithful behavior into the policy without introducing additional decoding operations at inference time.

AI USE STATEMENT

We used generative AI tools for checking syntactic and grammatical errors and improving the overall readability of the manuscript. We also used generative AI tools as evaluators to score model outputs in the experiments described in the paper. We carefully reviewed and verified all AI-assisted work and take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

We provide detailed information to facilitate reproduction of our results. Experimental datasets, models, and evaluation settings are provided in Section 4.1. Detailed training configurations and hyperparameters are provided in Appendix A, and the complete LLM-as-a-Judge prompt is provided in Appendix B. Pseudocode for CORE with GRPO and PPO is provided in Appendix D. An anonymous repository containing the source code is also provided here.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Baolong Bi, Shaohan Huang, Yiwei Wang, Tianchi Yang, Zihan Zhang, Haizhen Huang, Lingrui Mei, Junfeng Fang, Zehao Li, Furu Wei, et al. Context-dpo: Aligning language models for context-faithfulness. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10280–10300, 2025.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*, pp. 2924–2936, 2019.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

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

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *The Innovation*, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Yanwen Huang, Yong Zhang, Ning Cheng, Zhitao Li, Shaojun Wang, and Jing Xiao. Dynamic attention-guided context decoding for mitigating context faithfulness hallucinations in large language models. *arXiv preprint arXiv:2501.01059*, 2025.

Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, et al. Cladder: Assessing causal reasoning in language models. *Advances in Neural Information Processing Systems*, 36: 31038–31065, 2023.

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process reward models that think. *arXiv preprint arXiv:2504.16828*, 2025.

Wendi Li and Yixuan Li. Process reward model with q-value rankings. In *International Conference on Learning Representations*, volume 2025, pp. 14708–14726, 2025.

Yi-Chen Li, Tian Xu, Yang Yu, Xuqin Zhang, Xiong-Hui Chen, Zhongxiang Ling, Ningjing Chao, Lei Yuan, and Zhi-Hua Zhou. Generalist reward models: Found inside large language models. *arXiv preprint arXiv:2506.23235*, 2025.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The twelfth international conference on learning representations*, 2023.

Alisa Liu and Jiacheng Liu. The memotrap dataset, 2023.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*, 2020.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7052–7063, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Siya Qi, Rui Cao, Yulan He, and Zheng Yuan. Evaluating llms' assessment of mixed-context hallucination through the lens of summarization. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 16480–16503, 2025.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 2383–2392, 2016.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3505–3506, 2020.

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

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

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 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)*, pp. 783–791, 2024.

Shuzheng Si, Haozhe Zhao, Cheng Gao, Yuzhuo Bai, Zhitong Wang, Bofei Gao, Kangyang Luo, Wenhao Li, Yufei Huang, Gang Chen, et al. Teaching large language models to maintain contextual faithfulness via synthetic tasks and reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33001–33009, 2026.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Austin Xu, Srijan Bansal, Yifei Ming, Semih Yavuz, and Shafiq Joty. Does context matter? contextualjudgebench for evaluating llm-based judges in contextual settings. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9541–9564, 2025.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. Reclor: A reading comprehension dataset requiring logical reasoning. *arXiv preprint arXiv:2002.04326*, 2020.

Wenhao Yu, Meng Jiang, Peter Clark, and Ashish Sabharwal. Ifqa: A dataset for open-domain question answering under counterfactual presuppositions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8276–8288, 2023.

A Implementation Details

RL Algorithms We implement CORE on top of two mainstream RL paradigms, PPO and GRPO. For GRPO, we sample a group of $G = 8$ rollouts per input during training. All training procedures are conducted on a single node equipped with 8 NVIDIA A100 (80GB) GPUs, using DeepSpeed Rasley et al. (2020) ZeRO Stage 2 for memory-efficient distributed training.

Base Models All base models are downloaded from HuggingFace. For the Qwen3 series (Qwen3-1.7B, Qwen3-4B, Qwen3-8B), we enable thinking mode throughout both training and evaluation to leverage the model's built-in chain-of-thought reasoning capability. For Llama-3.1-8B-Instruct, we use the standard instruct variant.

Model Configurations Due to the substantial memory footprint of PPO's actor-critic architecture, we employ LoRA (Hu et al., 2022) with rank 64 and alpha 128 for models larger than 3B parameters (Qwen3-4B, Qwen3-8B, and Llama-3.1-8B-Instruct). For Qwen3-1.7B and all GRPO experiments, we perform full-parameter fine-tuning to maximize policy expressiveness.

Sequence Length During training, the maximum sequence length is set to 4096 tokens. During all evaluation, we increase the maximum generation length to 8192 tokens to accommodate longer reasoning chains on OOD benchmarks.

Correctness Verifier The accuracy reward r_{acc} is implemented as a rule-based verifier using strict string matching against the ground-truth answer. A response is considered correct only if the extracted answer string exactly matches the reference, ensuring a deterministic and reproducible binary signal without relying on any auxiliary model.

Context Masking The counterfactual context-masked forward pass is implemented by manipulating the attention mask (Vaswani et al., 2017) built into the language model. Specifically, we set the attention mask of all context tokens to zero, preventing the model from attending to the context while keeping the input sequence structure intact. This approach requires no modification to the model architecture and introduces minimal additional implementation complexity.

Hyperparameters Table 4 and Table 5 report the detailed hyperparameter configurations for all full-parameter training runs and LoRA-based PPO training runs, respectively. All models are trained for 1 epoch with the AdamW optimizer (Loshchilov & Hutter, 2017) in bfloat16 precision. We adopt a cosine learning rate schedule with a warmup ratio of 0.03.

Table 4: Hyperparameters of all GRPO training and PPO training for Qwen3-1.7B.

Config	Step1
Trainable parts	All Parameter
Global batch size	64
Batch size per GPU	8
Global learning rate	2e-6
DeepSpeed zero stage	2
Learning rate schedule	warmup + cosine decay
Warmup ratio	0.03
Weight decay	0
Epoch	1
Optimizer	AdamW
Precision	bfloat16
Model max length	4096

Table 5: Hyperparameters of PPO training for models larger than 3B parameters.

Config	Step1
Trainable parts	LoRA
LoRA Rank	64
LoRA Alpha	128
Global batch size	64
Batch size per GPU	8
Global learning rate	2e-6
DeepSpeed zero stage	2
Learning rate schedule	warmup + cosine decay
Warmup ratio	0.03
Weight decay	0
Epoch	1
Optimizer	AdamW
Precision	bfloat16
Model max length	4096

B LLM-AS-A-JUDGE PROMPT TEMPLATE

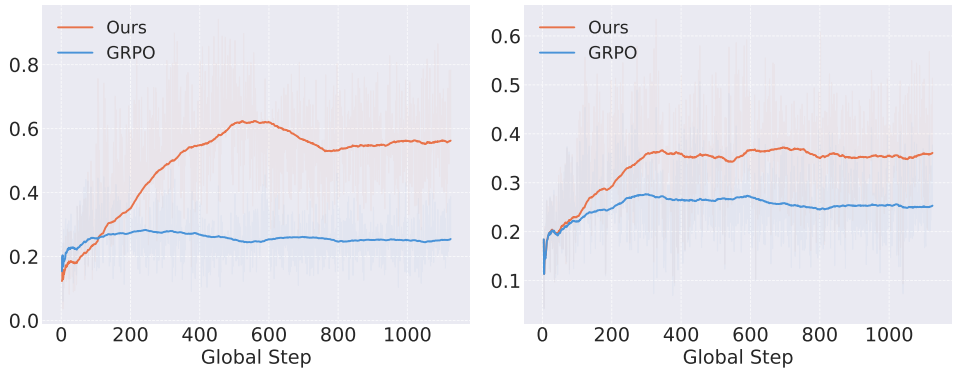
To rigorously evaluate the contextual faithfulness of the generated responses and detect knowledge conflicts, we employ the following prompt template for our LLM judge (Qwen3-235B-A22B-Instruct-2507). The prompt is explicitly designed to penalize contradictions rather than the mere use of external supplementary reasoning.

C MORE TRAINING DYNAMICS ANALYSIS

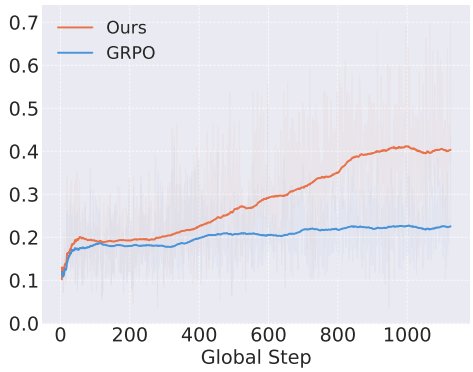
To complement the training dynamics analysis presented in Section 4.5, we provide CORE score curves for the remaining three base models in Figure 5. Across all model families and parameter

```
You are an expert AI evaluator assessing the "Contextual Faithfulness" of
     an AI assistant's response relative to a given context. Your goal is
     to identify if the response CONTRADICTS or MISREPRESENTS the
    information provided in the context.

### EVALUATION LOGIC:
1. FOCUS ON CONTRADICTION: Penalize the response heavily if it states
    something that directly conflicts with the facts, data, or logic in
    the Context.
2. TOLERATE SUPPLEMENTARY INFO: Do not penalize the response simply for
    using external common sense, basic math, or connecting logic, UNLESS
    that logic leads to a conclusion that contradicts the Context.
3. DATA INTEGRITY: If the Context provides specific numbers or facts, the
     Response must honor them. Changing a "5.5" from the Context into
    something else is a severe violation.

### Scoring Criteria (1 to 5 scale):
- Score 5 (No Conflict): The response is entirely consistent with the
    context. Any additional reasoning or common sense used does not
    contradict the provided information.
- Score 4 (Minor Inconsistency): The response is mostly consistent, but
    has a very slight imprecision or a minor leap in logic that doesn't
    significantly change the meaning.
- Score 3 (Partial Conflict): The response contains claims that go beyond
     the context and partially misrepresent its intent, though without a
    direct factual contradiction.
- Score 2 (Major Conflict): The response contains significant claims that
     directly contradict key facts or logic provided in the context.
- Score 1 (Severe Contradiction): The response is fundamentally at odds
    with the context, provides a completely different answer, or reverses
     the facts mentioned.

### Output Format:
Reason: [Identify any direct conflicts or confirm the consistency of key
    facts in one short sentence]
Score: \boxed{score}

### Input:
Please evaluate the faithfulness of the response.
[Context and Question]:{}
[Response]:{}
```

Figure 4: The strict evaluation prompt used for the LLM judge (Qwen3-235B-A22B-Instruct-2507) to score contextual faithfulness and penalize knowledge conflicts.



(a) Qwen3-4B (b) Qwen3-8B



(c) Meta-Llama-3.1-8B-Instruct

Figure 5: Contextual influence score curves of Ours and standard GRPO across three additional base models. Ours consistently exhibits higher contextual influence scores with an increasing trend during training, whereas the corresponding diagnostic scores of standard GRPO remain substantially lower across the evaluated model architectures.

scales, GRPO+CORE consistently exhibits a higher and steadily increasing CORE score throughout training, while standard GRPO stagnates at a significantly lower level. This universal pattern corroborates our central claim that standard outcome-based RL fails to internalize contextual faithfulness regardless of model architecture, whereas CORE reliably instills context-grounded reasoning as a stable behavioral trait.

D Training Algorithms Pseudocode

We present the detailed training procedures for CORE integrated with GRPO and PPO in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1 CORE with GRPO

Require: Policy π_θ, reference policy π_{ref}, dataset $\mathcal{D}$, reward coefficient α, group size G, KL coefficient β, normalization constant ϵ_{norm}, total epochs E

for epoch $= 1, \ldots, E$ **do**
  for each batch $(c, x, y^*) \sim \mathcal{D}$ **do**
    # Step 1: Rollout
    for $i = 1, \ldots, G$ **do**
      Sample $y_i = (y_1^i, \ldots, y_T^i) \sim \pi_\theta(\cdot \mid c, x)$
    end for
    # Step 2: Compute CORE reward
    for $i = 1, \ldots, G$ **do**
      Compute $r_{\text{acc}}^i = \mathbf{1}[y_i \text{ matches } y^*]$
      if $r_{\text{acc}}^i = 1$ **then**
        for $t = 1, \ldots, T$ **do**
          $\delta_t^i = \log \pi_\theta(y_t^i \mid c, x, y_{<t}^i) - \log \pi_\theta(y_t^i \mid x, y_{<t}^i)$
        end for
        $r_{\text{CORE}}^i = \frac{1}{T} \sum_{t=1}^{T} \delta_t^i$
      else
        $r_{\text{CORE}}^i = 0$
      end if
      $R_i = r_{\text{acc}}^i + \alpha \cdot r_{\text{CORE}}^i$
    end for
    # Step 3: Compute advantages
    for $i = 1, \ldots, G$ **do**
      $$\hat{A}_i = \frac{R_i - \text{mean}(\{R_1, \ldots, R_G\})}{\text{std}(\{R_1, \ldots, R_G\}) + \epsilon_{\text{norm}}}$$
    end for
    # Step 4: Update policy
    $\theta \leftarrow \theta + \nabla_\theta \left[\mathcal{L}_{\text{clip}}(\theta) - \beta \, \mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})\right]$
  end for
end for

Algorithm 2 CORE with PPO

Require: Policy π_θ, critic V_ϕ, reference policy π_{ref}, dataset $\mathcal{D}$, reward coefficient α, clip parameter ϵ, total epochs E

for epoch $= 1, \ldots, E$ **do**
  for each batch $(c, x, y^*) \sim \mathcal{D}$ **do**
    # Step 1: Rollout
    Sample $y = (y_1, \ldots, y_T) \sim \pi_\theta(\cdot \mid c, x)$
    # Step 2: Compute CORE reward
    Compute $r_{\text{acc}} = \mathbf{1}[y \text{ matches } y^*]$
    if $r_{\text{acc}} = 1$ **then**
      for $t = 1, \ldots, T$ **do**
        $\delta_t = \log \pi_\theta(y_t \mid c, x, y_{<t}) - \log \pi_\theta(y_t \mid x, y_{<t})$
      end for
      $r_{\text{CORE}} = \frac{1}{T} \sum_{t=1}^{T} \delta_t$
    else
      $r_{\text{CORE}} = 0$
    end if
    $R = r_{\text{acc}} + \alpha \cdot r_{\text{CORE}}$
    # Step 3: Compute advantage
    $\hat{A}_t = R - V_\phi(c, x, y_{<t})$
    # Step 4: Update critic
    $\phi \leftarrow \phi - \nabla_\phi \left(V_\phi(c, x, y_{<t}) - R\right)^2$
    # Step 5: Update policy
    $\theta \leftarrow \theta + \nabla_\theta \mathcal{L}_{\text{PPO}}(\theta)$
  end for
end for