Knowledge-Level Consistency Reinforcement Learning: Dual-Fact Alignment for Long-Form Factuality

Junliang Li **Yucheng Wang** **Yan Chen** **Yu Ran**
Ruiqing Zhang **Jing Liu*** **Hua Wu** **Haifeng Wang**
Baidu Inc.
{lijunliang03, chenyan22, liujing46}@baidu.com

Abstract

Hallucination in large language models during long-form generation remains difficult to address under existing reinforcement learning from human feedback frameworks, as their preference rewards often overlook the model's own knowledge boundaries. In this paper, we propose the **K**nowledge-**L**evel **C**onsistency Reinforcement Learning **F**ramework (**KLCF**), which re-examines this problem from a knowledge-consistency perspective. KLCF formalizes long-form factuality as a two-sided objective: maximizing coverage of the facts the base model can reliably express, while discouraging claims that fall beyond this boundary, thereby jointly optimizing recall and precision. Since parametric knowledge is latent and cannot be measured during online reinforcement learning, we anchor the boundary to the base model's *verified faithfully expressed knowledge*—facts that it expresses under a fixed sampling protocol and that external verification confirms. To achieve this, we design a Dual-Fact Alignment mechanism that approximates the recall term using a factual checklist constructed by sampling from the base model, and constrains hallucinations with a lightweight truthfulness reward model. Both components are jointly optimized and require no external retrieval throughout training. Experimental results demonstrate that KLCF consistently improves factuality metrics across multiple long-form benchmarks and model scales, effectively alleviating hallucination and over-conservatism while maintaining efficiency and scalability.

https://github.com/ki-ljl/KLCF
https://huggingface.co/collections/JunliangLi/klcf

1 Introduction

Large language models (LLMs) excel at tasks such as question answering, summarization, and complex reasoning [30, 55, 65, 18, 52, 44, 60]. However, hallucinations (generating content that conflicts with established facts) remain a critical threat to reliability [22, 49], especially in long-form generation where early errors can snowball [62, 24, 22]. Moreover, existing reinforcement learning from human feedback (RLHF) frameworks [2] rely on preference rewards but overlook the model's reliable knowledge boundary, inadvertently exacerbating the so-called "hallucination tax" [11, 45].

To evaluate and address the factuality and hallucination issues in long-form generation, a commonly employed approach is to decompose the model's output into atomic facts and verify them through external retrieval. The early works FActScore [36] and FacTool [9] established this paradigm. Subsequently, methods such as SAFE [53] and VeriScore [46] further refined the fact extraction and verification pipeline. Methods like FactTune-FS [47], FLAME [31], and FactAlign [21] use evaluators

*Corresponding Author. Contact: lijunliang.ki@gmail.com.

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

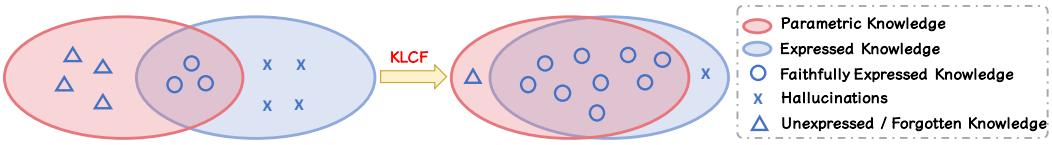


Figure 1: Knowledge-Level Consistency. Left: expressed knowledge both overflows the reliable boundary (×) and suppresses facts the model could express (△). Right: KLCF closes both gaps, growing the faithful overlap (○).

like FActScore to provide factuality alignment signals through external retrieval, successfully reducing hallucinations to some extent. However, most of these approaches are still confined to offline reinforcement learning (RL) scenarios. The low efficiency of external retrieval-based verification makes these methods difficult to apply to large-scale online RL. In terms of effectiveness, the main limitation of these approaches is a narrow focus on factual precision without considering recall, which often leads models to favor conservative responses. As illustrated in Fig. 1, the faithful overlap between the base model's parametric knowledge and the policy's expressed knowledge is not directly observable. Misalignment between the two degrades generation in opposite directions: under *expression overflow* the policy asserts claims beyond its reliable knowledge, which is the direct source of hallucination [15, 26], whereas under *knowledge suppression* it withholds facts it could reliably express, which is the conservatism induced by precision-only alignment [38, 32]. A faithful objective must therefore constrain both directions. Following the principle that LLMs' parametric knowledge originates from pre-training [39, 19], we propose that the core objective of long-form factuality alignment is to increase the **knowledge-level consistency** between the aligned model's "expressed knowledge" and the base model's "parametric knowledge". Since this overlap cannot be measured during online RL, we anchor the objective to an observable proxy: the base model's **verified faithfully expressed knowledge** $K_{\text{vfe}}(q)$, i.e., the claims that it expresses under a fixed sampling protocol and that external verification confirms as supported. Alignment then becomes two-sided: cover more of $K_{\text{vfe}}(q)$ (high recall), while discouraging claims that fall beyond it (high precision). We stress that $K_{\text{vfe}}(q)$ is a protocol-conditioned approximation of the base model's reliable knowledge boundary, not a prompt-independent estimate of its parametric knowledge. Prior work [25, 64] shows that this objective can be achieved by using internal knowledge as reward signals.

Building on the above theoretical motivation, we propose the **K**nowledge-**L**evel **C**onsistency Reinforcement Learning **F**ramework (**KLCF**). KLCF approximates this two-sided objective through a Dual-Fact Alignment mechanism: offline, it uses external retrieval to build a verifiable factual checklist from the base model's generations and trains a lightweight truthfulness reward model, thereby eliminating reliance on external retrieval during online RL. The checklist approximates the recall term via Monte Carlo estimation, while the truthfulness reward model approximates the precision term by scoring every generated claim, including those beyond the checklist. The two signals jointly optimize the precision and recall of expressed knowledge. Notably, although the reward signals are derived entirely from the base model's internal knowledge, the resulting model achieves consistent gains on external verification metrics such as FActScore and VeriScore, confirming that internal-knowledge optimization serves as a reliable proxy for external factuality evaluation. The overall reward design is lightweight, retrieval-free during training, and readily scalable to large-scale online RL. Our main contributions are as follows:

- We reframe long-form factuality alignment as a two-sided objective anchored to the base model's verified faithfully expressed knowledge, i.e., an observable, protocol-conditioned proxy for its reliable knowledge boundary.
- We introduce a Dual-Fact Alignment mechanism that jointly optimizes factual recall and precision through effective approximations of the alignment objective, forming a lightweight, retrieval-free reward system for scalable online RL.
- Extensive experiments across benchmarks, scales, and reasoning modes show consistent gains.

2 Knowledge-Level Consistency Reinforcement Learning

Fig. 2 summarizes KLCF against retrieval-based factuality alignment. We first formalize the two-sided objective and its base-anchored surrogate (Sections 2.1–2.2), then describe how the offline

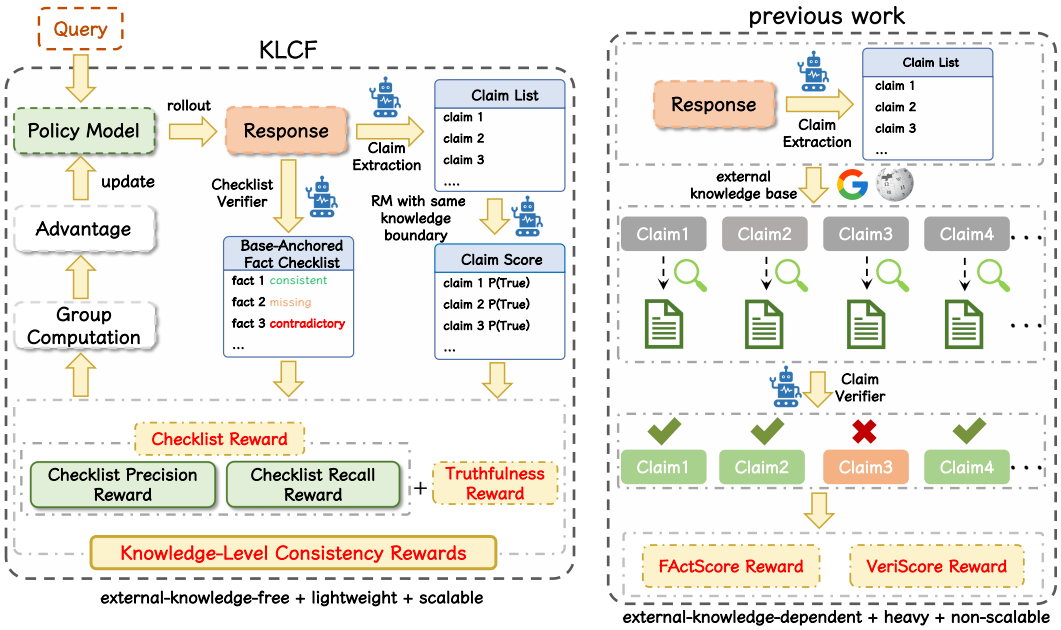


Figure 2: Comparison of KLCF (left) and retrieval-based factuality alignment (right). Prior methods issue one external search query per atomic claim at every training step; KLCF confines retrieval to a one-time offline stage that builds a factual checklist and a truthfulness reward model, leaving the online reward computation entirely local.

checklist and truthfulness reward model are constructed (Section 2.2.3), and finally how they are combined into the training reward and optimized (Section 2.3).

2.1 Problem Formulation

We formulate long-form factual generation as a latent *helpful-and-honest* objective. Given a query q, the policy model π_θ produces a response $y \sim \pi_\theta(\cdot \mid q)$, which is decomposed into a set of atomic claims $C(y) = \{c_1, c_2, \ldots\}$ by a claim extractor. Let $F^*(q)$ denote the latent set of all query-relevant and correct atomic facts that should ideally appear in a high-quality answer. We characterize factual helpfulness by the coverage of $F^*(q)$ and factual honesty by the truthfulness of the generated claims:

$$\text{Recall}^*(q, y) = \frac{1}{|F^*(q)|} \sum_{f \in F^*(q)} \mathbb{I}[f \in C(y)] \tag{1}$$

$$\text{Precision}^*(q, y) = \frac{1}{|C(y)|} \sum_{c \in C(y)} \mathbb{I}[\text{True}(c)] \tag{2}$$

The corresponding ideal utility is

$$U^*(q, y) = w_{\text{R}} \cdot \text{Recall}^*(q, y) + w_{\text{P}} \cdot \text{Precision}^*(q, y) + w_{\text{H}} \cdot H(q, y), \tag{3}$$

where $H(q, y)$ denotes the holistic quality of the response, including factors such as instruction following, coherence, and readability.

This formulation also clarifies the role of knowledge consistency. Let $K_{\text{para}}(q)$ denote the model's latent parametric knowledge and $K_{\text{expr}}(q)$ its expressed knowledge under a fixed prompting and sampling setup. Because $K_{\text{para}}(q)$ is not directly observable, the overlap $K_{\text{para}}(q) \cap K_{\text{expr}}(q)$ cannot be accessed exactly during online RL. In practice, we can only observe claims expressed by the model, and then estimate which part of that expressed knowledge is both correct and reliable.

Accordingly, we separate evaluation from training. Final evaluation is externally grounded and model-agnostic: we measure query-relevant factual recall, claim support rate under retrieval-based verification, and judge win rate for overall answer quality. Training, in contrast, must remain lightweight enough for online RL. We therefore optimize a retrieval-free surrogate reward anchored

to the base model's *verified faithfully expressed knowledge*. This surrogate is not identical to the final evaluation metric, but it is designed to be operationally aligned with it while remaining practical for large-scale RL.

2.2 Knowledge-Level Consistency Objective

Our core idea is to increase the faithful expression of knowledge that the base model can already express reliably, while suppressing claims that likely fall outside this reliable region. For each query q, we sample the base model multiple times under a fixed prompting protocol and extract its claims. Since the expressed claims can contain both correct and incorrect statements, we retain only those that are externally verified as supported facts. This yields the set of *verified faithfully expressed knowledge*:

$$K_{\mathrm{vfe}}(q) = \{f : f \in K_{\mathrm{expr}}^{\mathrm{base}}(q),\ \mathrm{Verify}(f) = \textsc{Support}\} \tag{4}$$

Operationally, $K_{\mathrm{vfe}}(q)$ is represented by a curated factual checklist $\Lambda(q)$ and by a truthfulness reward model trained on verified base-generated claims. Motivated by prior evidence [25, 58] that RL mainly elicits knowledge already acquired during pretraining, we use $K_{\mathrm{vfe}}(q)$ as an operational approximation to the base model's reliable knowledge boundary. This approximation is protocol-conditioned: writing it as $K_{\mathrm{vfe}}(q \mid \mathcal{C}, \mathcal{V})$ makes explicit its dependence on the sampling protocol $\mathcal{C}$ and the verification protocol $\mathcal{V}$, so a fact that the base model never expresses under $\mathcal{C}$, or that $\mathcal{V}$ fails to confirm, cannot enter $K_{\mathrm{vfe}}(q)$ even if it is encoded in $K_{\mathrm{para}}(q)$. It therefore constitutes a conservative finite sample of the reliable knowledge boundary rather than a prompt-independent estimate of parametric knowledge. Section 3.4 shows that the method is robust to the incompleteness and noise of this approximation.

2.2.1 Base-Anchored Surrogate Objective

Let $Q_{\mathrm{vfe}}(\cdot \mid q)$ denote the empirical distribution induced by the verified facts in $\Lambda(q)$, and let $s_{\mathrm{base}}(c) \in [0, 1]$ denote the truthfulness score assigned to claim c by the base-anchored reward model. We optimize the following surrogate objective:

$$\max_{\pi_\theta} \underbrace{\mathbb{E}_{f \sim Q_{\mathrm{vfe}}(\cdot|q)} \left[\Pr_{y \sim \pi_\theta(\cdot|q)} (f \in C(y)) \right]}_{\text{verified base knowledge recall}} + \lambda \underbrace{\mathbb{E}_{y \sim \pi_\theta(\cdot|q)} \left[\frac{1}{|C(y)| + \delta} \sum_{c \in C(y)} s_{\mathrm{base}}(c) \right]}_{\text{base-anchored truthfulness}}, \tag{5}$$

where $\lambda > 0$ balances the two terms and $\delta > 0$ avoids division by zero. The first term encourages the policy to cover more verified facts that the base model can faithfully express, thereby improving factual helpfulness. The second term suppresses unfaithful expression by rewarding claims that are judged likely to lie inside the base model's reliable knowledge boundary.

We refer to the joint optimization of these two terms as **Dual-Fact Alignment**. The terms are complementary rather than merely additive, since each rules out the other's degenerate solution: truthfulness alone is maximized by silence, which the recall term penalizes as missing coverage, while recall alone is maximized by indiscriminate assertion, which the truthfulness term penalizes claim by claim. Both terms are content-aware, so neither can be satisfied by simply saying more or saying less.

Importantly, Eq. (5) is a retrieval-free training surrogate rather than a direct estimator of the ideal objective in Eq. (3). Its purpose is to improve the faithful expression rate of relevant parametric knowledge using only signals that are available online during RL. We later validate this modeling choice empirically by showing that optimizing the base-anchored surrogate consistently improves externally grounded factual evaluation.

2.2.2 Practical Reward Instantiation

Eq. (5) is defined at the expectation level. During online RL, we instantiate it with single-sample rewards that can be computed from one generated response and the pre-built factual resources.

Checklist-scope recall. Given a response y and checklist $\Lambda(q)$, each checklist item is compared against the claims in y and labeled as CONSISTENT, CONTRADICTORY, or MISSING. Let $N_{\text{consistent}}$,

$N_{\text{contradictory}}$, and N_{missing} denote the corresponding counts. We define the **Fact Recall Reward** as

$$R_{\text{recall}}(y) = \frac{N_{\text{consistent}}}{N_{\text{consistent}} + N_{\text{contradictory}} + N_{\text{missing}}} \tag{6}$$

This reward measures how much of the verified checklist is faithfully covered by the response, and serves as a practical proxy for the verified base knowledge recall term in Eq. (5).

Checklist-scope precision. To sharpen supervision within the checklist scope, we additionally define the **Fact Precision Reward**

$$R_{\text{precision}}(y) = \begin{cases} \dfrac{N_{\text{consistent}}}{N_{\text{consistent}} + N_{\text{contradictory}}}, & \text{if } N_{\text{consistent}} + N_{\text{contradictory}} > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

This reward penalizes incorrect statements when the model attempts to cover checklist facts, and complements R_{recall} by discouraging inaccurate coverage.

Global truthfulness. Since the checklist only supervises facts that have already been mined from the base model, it cannot directly constrain content generated beyond the checklist scope. To address this, we train a lightweight **Truthfulness Reward Model** on verified base-generated claims (training details in Appendix D.1). For each atomic claim $c \in C(y)$, the model outputs a score $s_{\text{base}}(c) \in [0, 1]$, where higher scores indicate that the claim is more likely to fall inside the base model's reliable knowledge boundary. We define the **Truthfulness Reward** as

$$R_{\text{truth}}(y) = \begin{cases} \dfrac{1}{|C(y)|} \sum_{c \in C(y)} s_{\text{base}}(c), & \text{if } |C(y)| > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Checklist-aware truthfulness variant. We also consider a lower-noise variant that only scores claims not already covered by the verified checklist. Specifically, all items in $\Lambda(q)$ are concatenated into a pseudo-response, and the claims in $C(y)$ are verified against this pseudo-response using the same protocol as the checklist reward. Let $C_M(y)$ denote the subset of claims marked as MISSING. The variant reward is

$$R_{\text{truth}}^{\text{variant}}(y) = \begin{cases} 1, & \text{if } |C_M(y)| = 0, \\ \dfrac{1}{|C_M(y)|} \sum_{c \in C_M(y)} s_{\text{base}}(c), & \text{otherwise.} \end{cases} \tag{9}$$

Discussion. The rewards above form our knowledge-level consistency (KLC) reward family, the per-sample instantiation of Dual-Fact Alignment. R_{recall} increases the recall of verified base knowledge, $R_{\text{precision}}$ improves factual accuracy within the checklist scope, and R_{truth} or $R_{\text{truth}}^{\text{variant}}$ provides a global retrieval-free signal against unfaithful expression. Together, they approximate the two desiderata in Eq. (5) with per-sample signals that are efficient to compute during online RL, at a cost of $O(|\Lambda(q)| + |C(y)|)$ verifier and reward model calls per response and no retrieval.

2.2.3 Data Preparation for Factual Alignment

The checklist $\Lambda(q)$ and the truthfulness reward model both rely on a practical approximation to $K_{\text{vfe}}(q)$. We construct this approximation from three data sources, namely ELI5 [13], LongFact-Gen, and LongWiki-Gen, to ensure diversity and factual grounding. LongFact-Gen is regenerated using GPT-4.1 based on the original LongFact [53] prompts, and LongWiki-Gen is built by applying the methodology from [3] to the GoodWiki corpus [10], ensuring no overlap with their respective test sets. The overall pipeline, illustrated in Fig. 3, consists of the following four stages.

Stage 1: Sampling from the base model. To probe what the base model can express for a query q_i, we prompt it to generate multiple responses in a few-shot setting. Specifically, for each query we draw ν independent samples at a high temperature (in our experiments, $\nu = 8$), yielding responses $\{o_{i,1}, \ldots, o_{i,\nu}\}$ where $o_{i,j} \sim \pi_{\text{base}}(\cdot \mid q_i, \mathcal{C})$ and $\mathcal{C}$ denotes the few-shot context. This step expands the observable expressed knowledge of the base model under the chosen prompting protocol.

Stage 2: Claim extraction. We then extract verifiable atomic claims from the sampled responses using a lightweight claim extraction model f_{extract}. For each query q_i, we aggregate claims across

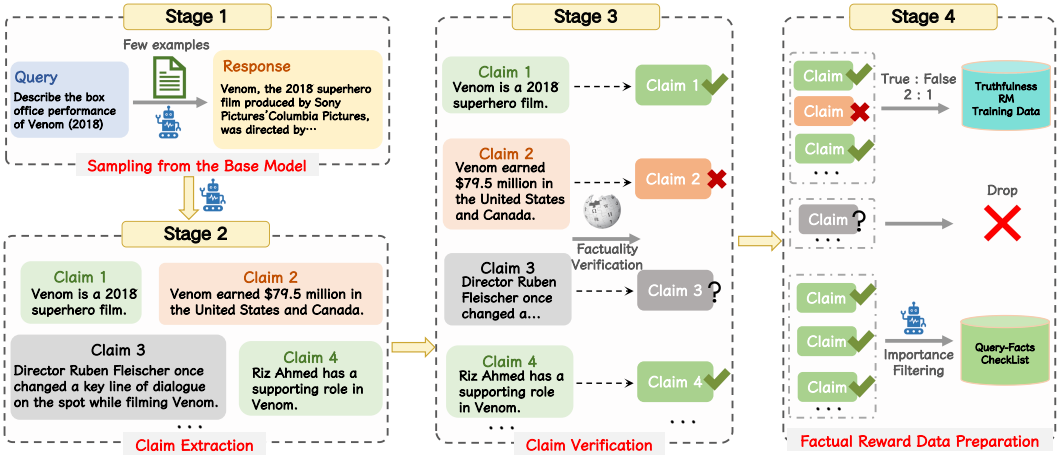


Figure 3: Offline data preparation pipeline. The process constructs the essential resources for knowledge-level consistency rewards—a factual checklist and truthfulness reward model training data—by extracting and verifying claims from the base model's responses.

all ν samples:

$$C(o_i) = \bigcup_{j=1}^{\nu} f_{\text{extract}}(o_{i,j}) = \{c_{i,1}, c_{i,2}, \ldots, c_{i,M_i}\} \tag{10}$$

This converts unstructured generations into a structured candidate fact set that can be externally verified.

Stage 3: External verification. Each extracted claim c is verified against a local Wiki20250716 knowledge index, built from a processed Wikipedia dump and indexed via [5, 48]. For each claim, the top-k relevant documents $D(c)$ are retrieved (with $k = 10$ in our experiments), and a verifier model f_{verify} assigns one of three labels:

$$l = f_{\text{verify}}(c, D(c)), \quad l \in \{\textsc{Support}, \textsc{Refute}, \textsc{Not Enough Info}\} \tag{11}$$

This step identifies which expressed claims are externally supported and therefore eligible to approximate the model's verified faithfully expressed knowledge.

Stage 4: Reward resource construction. Finally, the verified claims are curated into the two resources used during RL. First, the factual checklist $\Lambda(q_i)$ is built by aggregating all SUPPORT-labeled claims for query q_i and then filtering duplicates and low-importance items, resulting in a concise blueprint of verified base-expressible facts. Second, the truthfulness reward model training set is built from SUPPORT and REFUTE claims. The checklist provides direct supervision for checklist-based recall and precision rewards, while the reward model training set allows the truthfulness reward model to learn a retrieval-free approximation to the base model's reliable knowledge boundary.

2.3 Reward Design and Policy Optimization

We first formalize the key components involved in online RL. For a given query q_i, the policy model generates a structured response o_i = <think>$\mathcal{T}_i$ </think> <answer>$\mathcal{A}_i$ </answer>. Additionally, the factual checklist $\Lambda(q_i)$ is precomputed offline as described above. The final scalar reward is denoted by $R(o_i)$ for policy optimization.

2.3.1 Auxiliary Rewards

To preserve overall generation quality beyond factuality, we introduce three lightweight auxiliary rewards. **General Reward** $R_g(\mathcal{A}_i)$ is provided by a preference reward model to maintain readability and human-preferred answer quality. **Format Reward** $R_f(o_i)$ enforces the required <think> and <answer> structure for reliable parsing. **Length Penalty** $R_l(\mathcal{A}_i)$ discourages excessively long or redundant generations and promotes concise, information-dense answers. Full details of these auxiliary rewards are provided in Appendix C.5.

2.3.2 Final Reward

Our final reward combines the base-anchored KLC rewards with the auxiliary rewards:

$$R(o_i) = \kappa\, R_{\text{recall}}(\mathcal{A}_i) + \gamma\, R_{\text{precision}}(\mathcal{A}_i) + \mu\, \widetilde{R}_{\text{truth}}(\mathcal{A}_i) + 0.1\, R_g(\mathcal{A}_i) + R_l(\mathcal{A}_i) + R_f(o_i), \quad (12)$$

where $\widetilde{R}_{\text{truth}}$ denotes either R_{truth} or $R_{\text{truth}}^{\text{variant}}$, and $\kappa, \gamma, \mu \in [0, 1]$ satisfy $\kappa + \gamma + \mu = 1$. In this formulation, R_{recall} promotes the recall of verified base knowledge, $R_{\text{precision}}$ improves checklist-scope accuracy, and $\widetilde{R}_{\text{truth}}$ suppresses unfaithful claims beyond the checklist. Checklist-only, truth-only, and full factual reward variants are all obtained by setting the corresponding coefficients. We discuss the choice of κ, γ, and μ in Section 3.3, and describe the selection procedure in Appendix C.6.

2.3.3 Policy Optimization

To optimize the policy toward the reward objective above, we employ Group Relative Policy Optimization (GRPO) [44]. For a given query q, the behavior policy $\pi_{\theta_{\text{old}}}$ generates a group of G responses $\{o_i\}_{i=1}^{G}$. The advantage of the i-th response is computed by normalizing the group-level rewards $\{R(o_i)\}_{i=1}^{G}$:

$$\hat{A}_{i,t} = \frac{R(o_i) - \text{mean}(\{R(o_i)\}_{i=1}^{G})}{\text{std}(\{R(o_i)\}_{i=1}^{G})}, \qquad r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})} \quad (13)$$

When the within-group standard deviation is zero, we set $\hat{A}_{i,t} = 0$. The GRPO objective is then

$$\mathcal{J}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left(\min\Big(r_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_{i,t}\Big) - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})\right), \quad (14)$$

where ϵ is the clipping range, β controls the KL penalty, and π_{ref} is the reference model.

3 Experiments

3.1 Experimental Settings

Dataset and Metrics. We evaluate model performance on four long-form factual benchmarks: FActScore [36], HalluLens-LongWiki (abbreviated as LongWiki) [3], LongFact [53], and Factory [6]. Our metrics fall into two groups: claim-level factuality metrics obtained by retrieval-based verification (FActScore, Recall@K, Precision, F1@K), on which all our factuality claims depend; and WR (Win Rate), a pairwise GPT-4.1 preference judgment of overall answer quality, which we use only to check that factuality gains do not come at the cost of answer quality. Detailed descriptions of the training data processing pipeline are provided in Section 2.2.3 and Appendix C.1. For a detailed description of the evaluation metrics and related configuration settings, please refer to Appendix C.4.

Models and Baselines. To comprehensively evaluate the effectiveness of our proposed method and ensure a fair and thorough comparison across different training paradigms and model scales, we conduct extensive experiments on Qwen2.5 [56] family models of various sizes, including 7B, 14B, and 32B. Besides the pretrained base models (Base) and supervised fine-tuned models (DeepSeek Distillation Series [17]), our baselines span three routes to factuality alignment: inference-time verification (CoVe [12]), internal-signal-driven methods (Intuitor [66] and Self-Eval-P(True) [64], leveraging P(True) to refine factual precision), and external-knowledge-driven DPO [41] and GRPO [44] variants based on different reward signals (DPO + FActScore&KLC rewards, GRPO + FActScore&KLC rewards, and GRPO + FActScore + No. of Claims [7], which adds a claim-count term to counter conservatism and is the closest in motivation to ours).

Training Details. Training is conducted on 16×H100 (80GB) GPUs with full-parameter tuning. For all experiments, we employ a consistent learning rate of 1e-6 with cosine warmup. Further implementation details are provided in Appendix C.3.

3.2 Main Results

As summarized in Table 1, our method achieves significant improvements under both training paradigms. KLCF-zero, trained directly from the base model without SFT, attains a higher performance ceiling than SFT-based approaches on most metrics—we attribute this to avoiding the

Table 1: Results on Qwen2.5-14B. SFT denotes DeepSeek-R1-Distill-Qwen-14B. R@32 and Prec are Recall@32 and Precision. # Claims denotes the No. of Claims reward term. WR is a preference metric, not claim-level factuality. Best scores are bolded.

Model	FActScore	LongWiki			LongFact				Factory			
	FS	R@32	Prec	F1@32	R@64	Prec	F1@64	WR	R@64	Prec	F1@64	WR
Training from Base Model												
Base (10-shot)	46.8%	0.661	0.435	0.511	0.614	0.685	0.637	-	0.239	0.302	0.262	-
+ CoVe	46.0%	0.435	0.381	0.368	0.380	0.676	0.452	19.8%	0.179	0.304	0.206	17.9%
+ GRPO (FActScore)	54.8%	0.532	0.473	0.498	0.643	0.690	0.663	54.7%	0.253	0.318	0.278	35.9%
+ Self-Eval-P(True)	52.0%	0.498	**0.567**	0.504	0.232	**0.718**	0.338	8.6%	0.126	0.337	0.173	25.9%
+ KLCF-zero (Ours)	**61.2%**	**0.681**	0.552	**0.568**	**0.776**	0.704	**0.733**	**94.6%**	**0.296**	**0.350**	**0.309**	**65.5%**
Training from SFT Model												
SFT (Distill)	48.7%	0.503	0.532	0.494	0.403	0.760	0.507	-	0.138	0.286	0.175	-
+ Intuitor	52.7%	0.505	0.580	0.512	0.397	0.774	0.502	41.5%	0.145	0.294	0.183	41.8%
+ DPO (FActScore)	53.0%	0.502	0.538	0.496	0.407	0.765	0.512	47.2%	0.158	0.308	0.190	44.4%
+ DPO (KLC Rewards)	54.9%	0.505	0.584	0.517	0.432	0.779	0.530	56.9%	0.157	0.312	0.186	52.4%
+ GRPO (FActScore)	**57.6%**	0.472	**0.647**	0.510	0.406	0.780	0.504	29.6%	0.143	0.351	0.184	30.2%
+ GRPO (FActScore + # Claims)	53.9%	0.616	0.568	0.589	0.652	0.757	0.685	70.3%	0.213	0.361	**0.247**	61.5%
+ KLCF (Ours)	55.7%	**0.672**	0.638	**0.651**	**0.668**	**0.783**	**0.692**	**79.3%**	**0.221**	**0.383**	0.242	**66.5%**

knowledge forgetting and alignment tax introduced by SFT-based distillation [34, 14]. Unlike FActScore-based reward methods, which tend to trade recall for precision (most visibly on LongWiki), KLCF achieves the best F1 balance across benchmarks via Dual-Fact Alignment. Read separately from the claim-level metrics, WR corroborates that this balance matters: from SFT, GRPO + FActScore attains a high LongFact precision (0.780) yet wins only 29.6% of comparisons, whereas KLCF reaches 79.3% at comparable precision (0.783). Note also that the FActScore-based baselines are themselves retrieval-dependent, since computing their reward requires external search at every training step (Appendix D.4). Table 1 therefore already contrasts retrieval-based with retrieval-free training: trained from the base model, KLCF-zero outperforms GRPO + FActScore on every metric without issuing a single search query. Better elicitation of parametric knowledge can thus outperform external retrieval used as a reward signal, and KLCF remains orthogonal to retrieval augmentation at inference time. For a detailed analysis of reward trends and training dynamics, please refer to Appendix D.2.

3.3 Ablation Study

As shown in Table 2, ablating the Dual-Fact Alignment mechanism validates its critical role. Using only the Checklist Reward ($\kappa = 1/3, \gamma = 2/3, \mu = 0$) yields high recall but low precision, as excessive coverage introduces errors. Conversely, using only the Truthfulness Reward ($\kappa = 0, \gamma = 0, \mu = 1$) boosts precision at a severe cost to recall, indicating over-conservatism. Systematically adjusting the reward weights (from top to bottom in Table 2) reveals a clear trade-off: reducing the Checklist weight while increasing the Truthfulness weight monotonically improves precision but consistently degrades recall. This trend highlights the limitation of optimizing either objective in isolation. By integrating both rewards, KLCF achieves a superior balance. The Truthfulness Reward suppresses hallucinations encouraged by the Checklist Reward, while the Checklist Reward in turn counteracts the conservatism induced by the Truthfulness Reward, demonstrating the necessity of our dual mechanism.

3.4 Robustness to Checklist Quality

The factual checklist is a finite, protocol-conditioned approximation of the base model's verified faithfully expressed knowledge, so we examine whether KLCF depends on that approximation being complete or error-free. We perturb the checklist in two controlled ways and retrain: random deletion of entries, which simulates incompleteness, and replacement of entries with irrelevant claims from other queries, which simulates verification errors. Under the two most severe settings (deleting 50%

Table 2: Ablation study on reward components. Results are for Qwen2.5-14B under the RL-zero (training from base model) setting. The first row is the base model (10-shot) reference, and the gray row denotes our optimal reward weighting.

Reward Weight			**FActScore**	**LongWiki**			**LongFact**				**Factory**			
Recall κ	Precision γ	Truth μ	FS	R@32	Prec	F1@32	R@64	Prec	F1@64	WR	R@64	Prec	F1@64	WR
0	0	0	46.8%	0.661	0.435	0.511	0.614	0.685	0.637	-	0.239	0.302	0.262	-
$\frac{1}{3}$	$\frac{2}{3}$	0	48.8%	0.678	0.424	0.508	0.729	0.672	0.695	95.0%	0.243	0.266	0.248	**81.5%**
$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	55.7%	**0.695**	0.490	0.545	0.748	0.689	0.713	**96.4%**	0.273	0.314	0.290	79.3%
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	61.2%	0.681	0.552	**0.568**	0.776	0.704	0.733	94.6%	**0.296**	0.350	**0.309**	65.5%
0.4	0.2	0.4	57.5%	0.692	0.512	0.553	**0.781**	0.706	**0.737**	95.4%	0.281	0.323	0.292	70.3%
0	$\frac{1}{3}$	$\frac{2}{3}$	69.5%	0.434	0.679	0.464	0.642	0.761	0.685	49.2%	0.241	0.435	0.290	22.9%
0	0	1	**69.6%**	0.369	**0.724**	0.422	0.633	**0.793**	0.685	36.8%	0.243	**0.492**	0.296	15.3%

Table 3: Scaling study of KLCF on Qwen2.5-7B and 32B models under the "Training from Base Model" setting.

Model	**FActScore**	**LongWiki**			**LongFact**				**Factory**			
	FS	R@32	Prec	F1@32	R@64	Prec	F1@64	WR	R@64	Prec	F1@64	WR
Qwen2.5-7B												
Base (10-shot)	36.6%	0.529	0.359	0.412	0.522	0.600	0.548	-	0.203	0.267	0.223	-
+ CoVe	40.1%	0.293	0.357	0.294	0.326	0.605	0.383	15.5%	0.171	0.278	0.192	11.8%
+ KLCF-zero (Ours)	**55.0%**	**0.633**	**0.487**	**0.514**	**0.724**	**0.675**	**0.693**	**94.8%**	**0.258**	**0.301**	**0.271**	**78.9%**
Qwen2.5-32B												
Base (10-shot)	40.0%	0.664	0.453	0.520	0.531	0.691	0.575	-	0.234	0.318	0.256	-
+ CoVe	45.9%	0.375	0.410	0.355	0.379	0.718	0.466	19.5%	0.191	0.346	0.229	15.3%
+ KLCF-zero (Ours)	**59.3%**	**0.812**	**0.532**	**0.631**	**0.784**	**0.747**	**0.760**	**96.2%**	**0.307**	**0.434**	**0.322**	**71.3%**

or corrupting 20% of entries), LongFact F1@64 remains at 0.716 and 0.714, respectively. This still surpasses 0.663, the top baseline score from the base model. Enlarging the checklist in the opposite direction, by doubling the number of base-model samples, improves F1@64 by only 1.1%, indicating that our default setting already lies on a performance plateau. This tolerance is consistent with Dual-Fact Alignment: the truthfulness reward constrains claims globally and therefore offsets errors introduced through the checklist. Full results, together with an analysis of sensitivity to the sampling protocol, are provided in Appendix D.3.

3.5 Scaling Study

To verify the generalizability of the KLCF method across different model scales, we conduct scaling experiments on Qwen2.5-7B and 32B models. As shown in Table 3, for both the 7B and 32B models, our proposed method under the "Training from Base Model" setting significantly outperforms the original base model and the CoVe prompting approach across all claim-level factuality metrics. The results demonstrate that the performance improvement brought by KLCF is consistent regardless of model scale. Our method effectively enhances long-form factuality in small (7B), medium (14B), and large (32B) models, confirming that the framework possesses strong scalability and generalizability.

3.6 Generalization to Non-Thinking Models

While our primary experiments demonstrate the effectiveness of KLCF starting from a base model under a thinking-style output format, we further investigate its generalization capability by applying it to a standard non-thinking model. For this experiment, we adapt the reward function by removing the format reward (as it is unnecessary for a standard conversational model) and the general reward (relying instead on the KL divergence penalty to prevent excessive deviation), while retaining the length penalty to control verbosity. As shown in Table 4, KLCF still achieves a substantial improvement in claim-level factuality metrics, confirming that its effectiveness is not reliant on a thinking-style output format and generalizes robustly to conventional conversational models.

Table 4: Performance improvement on factuality benchmarks after adapting KLCF to a non-thinking, instruction-tuned model (Qwen2.5-14B-Instruct).

Model	FActScore	LongWiki			LongFact				Factory			
	FS	R@32	Prec	F1@32	R@64	Prec	F1@64	WR	R@64	Prec	F1@64	WR
Qwen2.5-14B-Instruct	61.1%	0.617	0.548	0.578	0.412	0.788	0.531	-	0.193	0.388	0.245	-
+ KLCF (Ours)	**65.0%**	**0.708**	**0.580**	**0.595**	**0.706**	**0.808**	**0.747**	**97.6%**	**0.290**	**0.398**	**0.306**	**67.1%**

3.7 Truthfulness Reward Analysis

We conduct a comparative analysis between the standard truthfulness reward calculation (Eq. (8)) and its variant (Eq. (9)) that leverages the pre-verified factual checklist to reduce evaluation noise. Both methods enhance factuality, but they exhibit distinct characteristics in balancing precision and recall. Due to space constraints, a detailed discussion of the results and their implications is provided in Appendix D.5.

3.8 Efficiency Analysis

To evaluate the efficiency of our reward design, we compare the computational cost of various factuality reward methods. Experimental details are provided in Appendix D.4. As summarized in Table 12 and Table 13, our KLC rewards achieve $3.46\times$ and $4.09\times$ speedups over FActScore and VeriScore, respectively, in serial mode, which further increase to $5.24\times$ and $5.40\times$ under parallel execution. More importantly, the detailed breakdown in Table 13 shows that our online reward computation takes only **127 seconds** per training step, compared to **1056 seconds** for FActScore—an **8.3**$\times$ reduction. While we incur a one-time offline preparation cost (20 hours) to construct checklists and train reward models, this investment is **permanently reusable** across multiple RL runs and model scales. In contrast, FActScore-based methods rely on external search APIs, which suffer from rate limits, network instability, and high latency.

3.9 Case Study

We compare Qwen2.5-14B-Instruct and its KLCF-trained version on a LongFact query (Table 15). The KLCF model gives a longer, more comprehensive response that draws on more of its internal knowledge, while containing fewer annotated errors, illustrating reduced hallucination without over-conservatism. The full query and annotated outputs are in Appendix D.6.

4 Conclusion, Limitations, and Future Work

We propose KLCF, a reinforcement learning framework that reduces long-form hallucination without inducing over-conservatism. It anchors a two-sided objective to the base model's verified faithfully expressed knowledge and realizes it through Dual-Fact Alignment, jointly optimizing recall and precision. KLCF surpasses existing baselines, and its retrieval-free reward design scales efficiently to large-scale online RL.

Limitations and Future Work. KLCF is evaluated only in closed-book long-form QA, the setting with mature claim-level protocols for both precision and Recall@K; summarization and dialogue tools mainly measure faithfulness and lack a comparable recall protocol. Transferring KLCF requires replacing only the checklist source and verifier grounding, plus extending these evaluation protocols. Factuality rewards are also applied only at the response level; step-wise alignment (e.g., sentence-level or chain-of-thought process rewards) is left to future work, as is long-form RAG, where recall can credit coverage of retrieved evidence and truthfulness can enforce faithfulness to it. Finally, anchoring to $K_{\text{vfe}}(q)$ has two consequences. It is protocol-conditioned: facts the base model never expresses under our sampling protocol, or that the verifier fails to confirm, never enter the reward, though Section 3.4 shows only mild degradation under checklist incompleteness and noise. It also introduces no knowledge beyond the base model's parameters, a deliberate trade-off favoring honest expression of learned knowledge; retrieval or human-annotated data can extend it when new knowledge is needed to update the knowledge base.

Acknowledgments and Disclosure of Funding

This work receives no external financial support. Computational resources are provided by Baidu Inc. to the research team. The authors declare that their employment at Baidu Inc. does not constitute dedicated funding for this project, and they have no competing interests to declare. The authors thank the NeurIPS reviewers for their valuable feedback, which helps improve the manuscript.

References

[1] S. Agarwal, Z. Zhang, L. Yuan, J. Han, and H. Peng. The unreasonable effectiveness of entropy minimization in llm reasoning. *arXiv preprint arXiv:2505.15134*, 2025.

[2] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[3] Y. Bang, Z. Ji, A. Schelten, A. Hartshorn, T. Fowler, C. Zhang, N. Cancedda, and P. Fung. Hallulens: Llm hallucination benchmark. *arXiv preprint arXiv:2504.17550*, 2025.

[4] J. A. Bishop, S. Ananiadou, and Q. Xie. Longdocfactscore: Evaluating the factuality of long document abstractive summarisation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10777–10789, 2024.

[5] D. Chen, A. Fisch, J. Weston, and A. Bordes. Reading Wikipedia to answer open-domain questions. In *Association for Computational Linguistics (ACL)*, 2017.

[6] M. Chen, Y. Li, X. Chen, A. Williams, G. Ghosh, and S. Yih. Factory: A challenging human-verified prompt set for long-form factuality. *arXiv preprint arXiv:2508.00109*, 2025.

[7] X. Chen, I. Kulikov, V.-P. Berges, B. Oğuz, R. Shao, G. Ghosh, J. Weston, and W.-t. Yih. Learning to reason for factuality. *arXiv preprint arXiv:2508.05618*, 2025.

[8] J. Cheng, T. Su, J. Yuan, G. He, J. Liu, X. Tao, J. Xie, and H. Li. Chain-of-thought prompting obscures hallucination cues in large language models: An empirical evaluation. *arXiv preprint arXiv:2506.17088*, 2025.

[9] I. Chern, S. Chern, S. Chen, W. Yuan, K. Feng, C. Zhou, J. He, G. Neubig, P. Liu, et al. Factool: Factuality detection in generative ai–a tool augmented framework for multi-task and multi-domain scenarios. *arXiv preprint arXiv:2307.13528*, 2023.

[10] E. Choi. Goodwiki dataset. `https://www.github.com/euirim/goodwiki`, September 2023.

[11] A. Cotra. Why ai alignment could be hard with modern deep learning. *Cold Takes*, 2021.

[12] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, and J. Weston. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*, 2023.

[13] A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, and M. Auli. Eli5: Long form question answering. *arXiv preprint arXiv:1907.09190*, 2019.

[14] T. Fu, D. Cai, L. Liu, S. Shi, and R. Yan. Disperse-then-merge: Pushing the limits of instruction tuning via alignment tax reduction. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2967–2985, Bangkok, Thailand, Aug. 2024. Association for Computational Linguistics.

[15] Z. Gekhman, G. Yona, R. Aharoni, M. Eyal, A. Feder, R. Reichart, and J. Herzig. Does fine-tuning llms on new knowledge encourage hallucinations? In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 7765–7784, 2024.

[16] Y. Gu, W. Zhang, C. Lyu, D. Lin, and K. Chen. Mask-dpo: Generalizable fine-grained factuality alignment of llms. *arXiv preprint arXiv:2503.02846*, 2025.

[17] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[18] A. Gupta, D. Singh, G. A. Cowan, N. Kadhiresan, S. Srivastava, Y. Sriraja, and Y. K. Mantri. Autosumm: A comprehensive framework for llm-based conversation summarization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 500–509, 2025.

[19] I. A. Haeun Yu, Pepa Atanasova. Revealing the parametric knowledge of language models: A unified framework for attribution methods. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2024.

[20] Y. He, S. Li, J. Liu, Y. Tan, W. Wang, H. Huang, X. Bu, H. Guo, C. Hu, B. Zheng, et al. Chinese simpleqa: A chinese factuality evaluation for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19182–19208, 2025.

[21] C.-W. Huang and Y.-N. Chen. Factalign: Long-form factuality alignment of large language models. *arXiv preprint arXiv:2410.01691*, 2024.

[22] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.

[23] T. Huang, S. Hu, F. Ilhan, S. F. Tekin, Z. Yahn, Y. Xu, and L. Liu. Safety tax: Safety alignment makes your large reasoning models less reasonable. *arXiv preprint arXiv:2503.00555*, 2025.

[24] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38, 2023.

[25] S. Kadavath, T. Conerly, A. Askell, T. Henighan, D. Drain, E. Perez, N. Schiefer, Z. Hatfield-Dodds, N. DasSarma, E. Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

[26] A. T. Kalai, O. Nachum, S. S. Vempala, and E. Zhang. Why language models hallucinate. *arXiv preprint arXiv:2509.04664*, 2025.

[27] J. Li and H. T. Ng. The hallucination dilemma: Factuality-aware reinforcement learning for large reasoning models. *arXiv preprint arXiv:2505.24630*, 2025.

[28] T. Li, Y. Zhang, P. Yu, S. Saha, D. Khashabi, J. Weston, J. Lanchantin, and T. Wang. Jointly reinforcing diversity and quality in language model generations. *arXiv preprint arXiv:2509.02534*, 2025.

[29] X. Li, T. Zhang, Y. Dubois, R. Taori, I. Gulrajani, C. Guestrin, P. Liang, and T. B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`, 5 2023.

[30] Z. Li, S. Fan, Y. Gu, X. Li, Z. Duan, B. Dong, N. Liu, and J. Wang. Flexkbqa: A flexible llm-powered framework for few-shot knowledge base question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 18608–18616, 2024.

[31] S.-C. Lin, L. Gao, B. Oguz, W. Xiong, J. Lin, W.-t. Yih, and X. Chen. Flame: Factuality-aware alignment for large language models. *Advances in Neural Information Processing Systems*, 37:115588–115614, 2024.

[32] Y. Lin, H. Lin, W. Xiong, S. Diao, J. Liu, J. Zhang, R. Pan, H. Wang, W. Hu, H. Zhang, et al. Mitigating the alignment tax of rlhf. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 580–606, 2024.

[33] C. Y. Liu, L. Zeng, Y. Xiao, J. He, J. Liu, C. Wang, R. Yan, W. Shen, F. Zhang, J. Xu, Y. Liu, and Y. Zhou. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *arXiv preprint arXiv:2507.01352*, 2025.

[34] Y. Luo, Z. Yang, F. Meng, Y. Li, J. Zhou, and Y. Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

[35] D. Metropolitansky and J. Larson. Towards effective extraction and evaluation of factual claims. *arXiv preprint arXiv:2502.10855*, 2025.

[36] S. Min, K. Krishna, X. Lyu, M. Lewis, W.-t. Yih, P. W. Koh, M. Iyyer, L. Zettlemoyer, and H. Hajishirzi. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*, 2023.

[37] J. Ni, M. Shi, D. Stammbach, M. Sachan, E. Ash, and M. Leippold. Afacta: Assisting the annotation of factual claim detection with reliable llm annotators. *arXiv preprint arXiv:2402.11073*, 2024.

[38] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[39] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller. Language models as knowledge bases? In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[40] M. Prabhudesai, L. Chen, A. Ippoliti, K. Fragkiadaki, H. Liu, and D. Pathak. Maximizing confidence alone improves reasoning. *arXiv preprint arXiv:2505.22660*, 2025.

[41] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.
[42] B. Ren, S. Qiao, W. Yu, H. Chen, and N. Zhang. Knowrl: Exploring knowledgeable reinforcement learning for factuality. *arXiv preprint arXiv:2506.19807*, 2025.
[43] S. Shafayat, F. Tajwar, R. Salakhutdinov, J. Schneider, and A. Zanette. Can large reasoning models self-train? *arXiv preprint arXiv:2505.21444*, 2025.
[44] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
[45] M. Sharma, M. Tong, T. Korbak, D. Duvenaud, A. Askell, S. R. Bowman, N. Cheng, E. Durmus, Z. Hatfield-Dodds, S. R. Johnston, et al. Towards understanding sycophancy in language models. *arXiv preprint arXiv:2310.13548*, 2023.
[46] Y. Song, Y. Kim, and M. Iyyer. Veriscore: Evaluating the factuality of verifiable claims in long-form text generation. *arXiv preprint arXiv:2406.19276*, 2024.
[47] K. Tian, E. Mitchell, H. Yao, C. Manning, and C. Finn. Fine-tuning language models for factuality. *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023.
[48] L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, and F. Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.
[49] Y. Wang, M. Wang, M. A. Manzoor, F. Liu, G. Georgiev, R. J. Das, and P. Nakov. Factuality of large language models in the year 2024. *CoRR*, 2024.
[50] Z. Wang, J. Zeng, O. Delalleau, D. Egert, E. Evans, H.-C. Shin, F. Soares, Y. Dong, and O. Kuchaiev. Helpsteer3: Human-annotated feedback and edit data to empower inference-time scaling in open-ended general-domain tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25640–25662, 2025.
[51] M. Wanner, B. Van Durme, and M. Dredze. Dndscore: Decontextualization and decomposition for factuality verification in long-form text generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 23620–23637, 2025.
[52] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
[53] J. Wei, C. Yang, X. Song, Y. Lu, N. Hu, J. Huang, D. Tran, D. Peng, R. Liu, D. Huang, et al. Long-form factuality in large language models. *Advances in Neural Information Processing Systems*, 37:80756–80827, 2024.
[54] Z. Wei, X. Yang, K. Sun, J. Wang, R. Shao, S. Chen, M. Kachuee, T. Gollapudi, T. Liao, N. Scheffer, et al. Truthrl: Incentivizing truthful llms via reinforcement learning. *arXiv preprint arXiv:2509.25760*, 2025.
[55] Y. Xu, S. He, J. Chen, Z. Wang, Y. Song, H. Tong, G. Liu, K. Liu, and J. Zhao. Generate-on-graph: Treat llm as both agent and kg in incomplete knowledge graph question answering. *arXiv preprint arXiv:2404.14741*, 2024.
[56] Q. A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, G. Dong, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y.-C. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, Z. Qiu, S. Quan, and Z. Wang. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024.
[57] Q. Yu, Z. Zhang, R. Zhu, Y. Yuan, X. Zuo, Y. Yue, W. Dai, T. Fan, G. Liu, L. Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.
[58] Y. Yue, Z. Chen, R. Lu, A. Zhao, Z. Wang, S. Song, and G. Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.
[59] C. Zhang, X. Zhu, C. Li, N. Collier, and A. Vlachos. Reinforcement learning for better verbalized confidence in long-form generation. *arXiv preprint arXiv:2505.23912*, 2025.
[60] J. Zhang, X. Wang, W. Ren, L. Jiang, D. Wang, and K. Liu. Ratt: A thought structure for coherent and correct llm reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26733–26741, 2025.
[61] K. Zhang, Q. Yao, S. Liu, Y. Wang, B. Lai, J. Ye, M. Song, and D. Tao. Consistent paths lead to truth: Self-rewarding reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.08745*, 2025.

[62] M. Zhang, O. Press, W. Merrill, A. Liu, and N. A. Smith. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*, 2023.
[63] Q. Zhang, H. Wu, C. Zhang, P. Zhao, and Y. Bian. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. *arXiv preprint arXiv:2504.05812*, 2025.
[64] X. Zhang, B. Peng, Y. Tian, J. Zhou, L. Jin, L. Song, H. Mi, and H. Meng. Self-alignment for factuality: Mitigating hallucinations in llms via self-evaluation. *arXiv preprint arXiv:2402.09267*, 2024.
[65] Y. Zhang, H. Jin, D. Meng, J. Wang, and J. Tan. A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods. *arXiv preprint arXiv:2403.02901*, 2024.
[66] X. Zhao, Z. Kang, A. Feng, S. Levine, and D. Song. Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*, 2025.
[67] Y. Zuo, K. Zhang, L. Sheng, S. Qu, G. Cui, X. Zhu, H. Li, Y. Zhang, X. Long, E. Hua, et al. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*, 2025.

Appendices

A Supplementary Statements **15**

A.1 Statement on LLM Usage 15

A.2 Ethics Statement 16

B Related Work **16**

B.1 Long-Form Factuality Evaluation 16

B.2 Long-Form Factuality Alignment 16

B.3 RL Without External Feedback 17

B.4 Alignment Tax 17

C Implementation Details **18**

C.1 Data Construction 18

C.2 Baselines 19

C.3 Training Setups 19

C.4 Evaluation Setups 19

C.5 Auxiliary Rewards 21

C.6 Reward Weight Selection 22

D Additional Experiments **22**

D.1 Verifier Training 22

D.2 Main Results 23

D.3 Checklist Quality and Robustness 25

D.4 Efficiency Analysis 27

D.5 Truthfulness Reward Analysis 28

D.6 Case Study 29

E Prompts **31**

E.1 Data Construction 31

E.2 Training 33

E.3 Evaluation 35

A Supplementary Statements

A.1 Statement on LLM Usage

This paper employed an LLM (GPT-5 by OpenAI) in a limited capacity, strictly as a writing aid. The model's role was confined to proofreading and polishing select passages to enhance readability and language fluency. Specifically, it was used to rephrase sentences for better flow and correct minor grammatical errors in introductory, explanatory, and concluding sections. Crucially, the LLM did not contribute to the core research process. It was not used for generating ideas, designing experiments, conducting analyses, or interpreting results. The fundamental scientific contributions and the intellectual substance of the paper are solely the product of the authors' work. The LLM served only as a passive tool to assist in the final presentation of the manuscript.

A.2 Ethics Statement

This work presents a reinforcement learning framework aimed at improving the factuality of LLMs in long-form generation, with the primary goal of mitigating harmful hallucinations and enhancing the reliability of AI-generated content. Our research utilizes publicly available datasets (e.g., ELI5) or synthetically generated data based on publicly available sources (e.g., Wikipedia), which do not contain private or sensitive personal information. We acknowledge that while our work seeks to reduce the generation of incorrect information, the potential for LLMs to be misused for generating convincing but false content remains a broader societal challenge. We have conducted our research with a commitment to the Code of Ethics, focusing on a beneficial application of AI. To the best of our knowledge, this work does not present any immediate, direct, or novel risks to privacy, security, or fairness beyond those common to the field of LLM development. We fully adhere to the Code of Ethics and affirm that there are no potential conflicts of interest to disclose.

B Related Work

B.1 Long-Form Factuality Evaluation

Fine-grained factuality assessment of generated text serves as a fundamental prerequisite for reliable factuality alignment. Early approaches typically decompose long-form text into atomic facts and verify them against external knowledge to evaluate factual accuracy. FActScore [36] and FacTool [9] establish this paradigm, employing retrieval and large verification models to determine the veracity of each atomic fact. Subsequently, methods such as SAFE [53] and VeriScore [46] further refine the fact extraction and verification pipeline, while numerous follow-up studies [37, 4, 51] continue to advance this direction by improving evaluation effectiveness and efficiency. However, these methods generally rely on external retrieval interfaces and large-scale verification models, resulting in high computational overhead and making them unsuitable for training scenarios in online reinforcement learning that require high-frequency reward signals. Beyond the pipeline itself, the scope of these metrics has also evolved. FActScore [36] defines factuality as precision alone, measuring the proportion of supported atomic facts without any notion of coverage; LongFact [53] subsequently introduced Recall@K and F1@K, and LongWiki [3] further refined them. Training reward design has lagged behind this shift: evaluation now treats factuality as a precision–recall balance, yet the signals used to optimize it remain one-sided, a gap that directly motivates our dual-fact formulation.

B.2 Long-Form Factuality Alignment

To directly enhance the factuality of LLMs, researchers have proposed various alignment methods. Among approaches based on supervised fine-tuning (SFT) and direct preference optimization (DPO), methods such as FactTune-FS [47] and FLAME [31] leverage scores from evaluators like FActScore to guide models toward factual preferences, while FactAlign [21] and Mask-DPO [16] construct preference pairs based on external retrieval for DPO training. However, these methods are generally confined to offline training scenarios and often prioritize factual precision at the expense of recall, leading to overly conservative model behavior.

In recent years, a growing body of work has begun incorporating factuality signals as rewards within RL frameworks. For example, KnowRL [42] introduces factuality rewards based on knowledge verification during training to help the model recognize its knowledge boundaries, and TruthRL [54] proposes a triple reward mechanism that distinguishes between correct answers, hallucinations, and abstentions, encouraging the model to refrain from responding when uncertain to improve truthfulness. Additionally, several other studies [7, 59, 27] explore the use of various types of factuality signals as rewards to enhance model capability and reduce hallucinations. Nevertheless, most of these methods still rely on time-consuming external retrieval processes, which limits their applicability in large-scale online RL training. In contrast to the above works, KLCF adopts a lightweight reward mechanism that requires no external retrieval during online training. Through Dual-Fact Alignment, KLCF jointly optimizes factual recall and precision, effectively enhancing factuality while avoiding the decline in expressiveness caused by excessive conservatism.

The conservatism these methods exhibit is intrinsic to a precision-only reward. Precision can be raised either by generating more correct facts or by simply emitting fewer claims, and the latter is the path of least resistance, since dropping low-confidence claims removes errors faster than it removes

supported facts. The shift from offline preference optimization to online RL then determines how far this shortcut is exploited. Under DPO the policy cannot iteratively search for reward loopholes, so conservatism bias stays bounded; under online RL, "shorten the answer and keep only high-confidence claims" becomes a stable optimum. Table 1 shows this with the same FActScore signal under both paradigms: DPO barely changes either precision or recall, whereas online GRPO raises precision substantially while recall stagnates or declines. An explicit recall signal is therefore not merely a performance refinement but a necessary component for preventing precision-only reward hacking. Among prior designs, No. of Claims [7] is the one that recognizes this conservatism problem, but it proxies recall by claim count, which does not distinguish which content is expressed and thus admits the opposite failure of rewarding redundant assertion; its reward also still depends on online retrieval. KLCF instead makes both sides content-aware: coverage is credited only for specific verified facts, and truthfulness is assessed claim by claim.

B.3 RL Without External Feedback

To overcome the limitations of verifiers, researchers are increasingly turning to reward signal generation methods that operate without external feedback. These label-free RL approaches primarily develop along two representative directions. One line of research focuses on deriving reward signals from models' internal confidence estimates, enhancing predictive certainty by rewarding low-entropy or self-consistent outputs [66, 61, 43, 1, 28, 40]. For instance, Intuitor [66] leverages the model's own confidence estimates as unsupervised optimization signals to effectively improve the factuality of generated content. The other predominant paradigm constructs supervisory signals through collective decision-making mechanisms [63, 67], as exemplified by TTRL [67], which aggregates majority-voted answers from multiple generated samples to serve as pseudo-ground-truth labels for reinforcement learning updates. KLCF shares the retrieval-free property of this line of work, but differs in what its endogenous signal is anchored to: rather than to the model's own confidence, which is itself distorted during alignment and therefore risks a self-reinforcing bias, our reward is anchored to base-model knowledge that has been externally verified offline.

B.4 Alignment Tax

The alignment tax is a prevalent phenomenon in RLHF and SFT processes. Some studies [38, 23, 32] indicate that alignment often leads models to become "overly conservative", sacrificing capability for improved safety. Although some models, such as the DeepSeek-R1-Distill series, perform well on general instruction-following tasks, their performance on long-form factuality benchmarks often falls short of that of the original base models, which serves as a concrete manifestation of the alignment tax (see Table 1). More critically, current FActScore-guided RL methods further exacerbate this issue: to avoid negative judgments from external verifiers, models adopt a "less-is-better" generation strategy, leading to a substantial decline in recall and ultimately reducing the overall F1-score. Therefore, a key motivation of this work is to mitigate the alignment tax by conducting RL directly from the base model with knowledge-level consistency as the objective. This approach enables the model to fully express its parametric knowledge while avoiding content that exceeds its knowledge boundaries.

Recent work supplies both analytical and before/after empirical grounding for this phenomenon. Kalai et al. [26] reduce generation errors to an Is-It-Valid classification problem and show that binary scoring in post-training rewards guessing while penalizing abstention, so the optimal aligned strategy is to answer confidently under uncertainty—an analytical account of how alignment pressure drives the policy away from the base model's calibrated knowledge. Consistent with this, as reported in [26], GPT-4 is well calibrated before RLHF but noticeably miscalibrated afterwards, with probability mass concentrating at the high-confidence end. In the opposite direction, Chinese-SimpleQA [20] reports cross-family before/after comparisons in which factual scores drop substantially after SFT/RLHF, indicating that alignment can systematically suppress the expression of knowledge a model already holds. Gekhman et al. [15] complete the picture from the other side, showing that fine-tuning a model on knowledge beyond its pre-training boundary measurably increases hallucination. Together, these results describe the two directions of misalignment that KLCF targets: expression beyond the reliable knowledge boundary, and suppression of knowledge within it.

C Implementation Details

C.1 Data Construction

RL Training Dataset. ELI5 [13] is a publicly available long-form question-answering dataset, from which we select 7993 samples for training. To avoid overlap with evaluation benchmarks, we construct two new datasets adhering to established methodologies: LongFact-Gen, containing 4348 samples generated by reproducing the original LongFact [53] prompt strategy using GPT-4.1; and LongWiki-Gen, comprising 2248 samples built from the GoodWiki [10] corpus following the procedure described in HalluLens [3]. These training sets are distinct from test benchmarks.

We then follow the data preprocessing pipeline outlined in Section 2.2.3 to perform uniform cleaning and formatting on all three datasets. The training datasets are summarized in Table 5.

Table 5: Statistics of training data.

	RL			DPO
	Dataset Size	**Avg. True Claims**	**Avg. Checklist Entries**	**Dataset Size**
7B	12680	34.76	10.09	12680
14B	13230	38.34	11.03	13230
32B	12967	37.19	11.10	12967

DPO Training Dataset. In this work, we employ the DeepSeek-R1-Distill-Qwen models for DPO training. To construct the DPO training dataset, we first sample 8 responses for each query from the RL dataset using models of three sizes (7B, 14B, and 32B). Subsequently, we extract claims from all responses for every query using a lightweight trained claim extraction model along with Prompt 2. We then verify all claims of each response against a locally built Wiki20250716 index, utilizing Qwen2.5-72B-Instruct and Prompt 3, to obtain the FActScore for every response. Finally, we form the <chosen, rejected> pairs by selecting the highest- and lowest-scoring responses for each query, thereby completing the preparation of the DPO dataset.

Similarly, to construct DPO training data based on the reward mechanism proposed in this paper, we need to assign a reward value to each response generated for every query, according to the reward computation method defined in Section 2.2.2. Specifically, we first use a lightweight checklist verification model with Prompt 7 to extract the checklist verification results for each response, and then compute the reward R_{recall} and the reward $R_{\text{precision}}$ based on Eq. (6) and Eq. (7), respectively. Subsequently, based on the previously extracted claims, we employ a truthfulness reward model with Prompt 6 to obtain the credibility probability score for each claim, and calculate the reward R_{truth} according to Eq. (8). Then, the final reward value for each response is computed using only the factual terms of Eq. (12), i.e., $\kappa R_{\text{recall}} + \gamma R_{\text{precision}} + \mu R_{\text{truth}}$, as the auxiliary rewards do not apply to offline preference pair construction. Finally, for each query, we select the responses with the highest and lowest reward values to form the <chosen, rejected> preference pairs. The training datasets used for DPO are summarized in Table 5.

Test Dataset. In this paper, we conduct a comprehensive evaluation of the trained models on four public long-form text evaluation benchmarks: FActScore [36], HalluLens-LongWiki [3], LongFact [53], and Factory [6]. The statistics for these datasets are detailed in Table 6. Specifically, for FActScore, we use 500 samples provided in its official release as the test set. For HalluLens-LongWiki, we strictly follow the methodology described in the original paper to generate 250 test samples. For the LongFact benchmark, we select 250 samples from its "Objects" subset. For the Factory benchmark, we randomly select 250 samples from its "Hard" subset to form the test set.

Table 6: Statistics of test datasets.

	FActScore	**HalluLens-LongWiki**	**LongFact-Objects**	**Factory-Hard**
Total Dataset Size	500	N/A	1140	421
Test Dataset Size	500	250	250	250

C.2 Baselines

This section provides a detailed description of the baseline methods included in our experiments to ensure a comprehensive and fair comparison. Our evaluation encompasses a diverse set of representative approaches across different training paradigms.

The **Base** model refers to the pretrained Qwen2.5 models (7B, 14B, 32B) evaluated in a 10-shot setting, serving as the foundational performance benchmark. The prompting-based baseline is represented by **CoVe** [12], which reduces hallucinations through a self-verification mechanism during inference without updating model parameters. For supervised fine-tuning, we include the **DeepSeek Distillation Series** [17]. We also compare against **Intuitor** [66], an unsupervised method that improves factuality by leveraging the model's intrinsic confidence estimates. We further include **Self-Eval-P(True)** [64], which likewise relies only on internal signals: the model scores the truthfulness of its own outputs by its probability of judging them as True, and this self-evaluation is used as the training signal to improve factual precision without external retrieval.

Furthermore, we incorporate several reinforcement learning baselines to isolate the contribution of our reward design. These include DPO and GRPO optimized with different reward signals: **DPO + FActScore** and **GRPO + FActScore** use the external FActScore metric as the reward signal, favoring precision but often at the cost of recall. **GRPO + FActScore + No. of Claims** [7] augments the FActScore reward with a term proportional to the number of generated claims, explicitly countering the tendency of precision-only rewards to shorten answers; it is the baseline closest in motivation to ours. However, claim count does not distinguish which content is expressed, so it can reward redundant assertions, and its reward still requires online retrieval at every training step. In contrast, **DPO + KLC Rewards** and **GRPO + KLC Rewards** (the latter being our full KLCF method) are trained using the knowledge-level consistency rewards introduced in this work, enabling joint optimization of recall and precision through Dual-Fact Alignment. Recent factuality RL methods such as KnowRL [42] and TruthRL [54] instead target short-form single-answer QA and provide no coverage incentive, so they do not transfer to long-form multi-fact generation; their long-form counterparts are represented by the FActScore-based baselines above.

C.3 Training Setups

This section summarizes the main hyperparameters of each training stage; settings are kept uniform within a stage, and the complete configuration files are released with our code (linked in the abstract).

RL. We use VeRL with full-parameter training on 2 nodes of 8 $\times$ H100 (80GB) GPUs. Each step samples 64 prompts with $G = 8$ rollouts each (temperature 1.0, top-p 1.0), with maximum prompt and response lengths of 8192 and 4096 tokens. We train for 2 epochs with learning rate 1×10^{-6}, weight decay 0.1, and a cosine schedule with 10% warmup. No KL loss or entropy bonus is used ($\beta = 0$). The length penalty uses $L_m = 2048$ and $L_m - L_c = 850$. Intuitor uses the same settings except for a KL loss coefficient of 0.05, following its original paper.

DPO. We use LLaMA-Factory with a sigmoid preference loss ($\beta = 0.1$), cutoff length 4096, per-device batch size 1 with 8 gradient accumulation steps, and a cosine schedule with 10% warmup, for 2 epochs on the preference datasets.

Verifier and reward model SFT. The checklist verifier and claim extractor are fine-tuned with VeRL on 8 H100 GPUs for 3 epochs, with batch size 64, maximum length 12288, learning rate 1×10^{-6}, AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 0.01), gradient clipping 1.0, and a cosine schedule with 10% warmup. The truthfulness reward model uses the same settings except for a maximum length of 4096 and a learning rate of 1×10^{-5}.

C.4 Evaluation Setups

C.4.1 Setups

In this study, we adopt corresponding metrics for different evaluation benchmarks. For FActScore [36], we directly use the FActScore metric proposed in the original paper, calculating the scores for each model on the 500 officially released test samples. For the 250 samples from HalluLens-LongWiki [3], we likewise employ the three metrics recommended by the original paper: Recall@32, Precision, and F1@32. For the LongFact and Factory datasets, we adopt the evaluation pipeline proposed in

VeriScore [46] and report three key metrics: Recall@64, Precision, and F1@64. Additionally, for the LongFact and Factory datasets, we introduce the Win Rate (WR) metric to provide a complementary performance perspective.

FActScore. The automated FActScore evaluation in the original paper's code[2] typically relies on models such as ChatGPT for atomic fact decomposition and uses the outdated Wikipedia dump it provides for judgment. However, due to the high cost of ChatGPT and the outdated nature of the original Wiki knowledge base, we make adjustments to this evaluation pipeline.

To address these issues, our approach adopts a more cost-effective and timely solution. Specifically, we use the powerful open-source model Qwen2.5-72B-Instruct to replace ChatGPT for decomposing text into atomic facts. In addition, we build a local Wiki20250716 index to serve as the latest knowledge base. Based on the information retrieved from this knowledge base, we also use the Qwen2.5-72B-Instruct model to judge the veracity of each atomic fact. This pipeline ensures the evaluation's cost-effectiveness, data timeliness, and result reliability.

HalluLens-LongWiki. For the HalluLens-LongWiki[3] benchmark, we follow its standard factuality framework of claim extraction, retrieval, and verification. While we use the benchmark's official knowledge base for retrieval, we employ Qwen2.5-72B-Instruct for the extraction and verification steps to ensure a cost-effective and reproducible setup.

LongFact. For the LongFact[4] benchmark, we randomly sample 250 instances from its more challenging "Objects" subset to form the test set. The SAFE framework proposed in LongFact is a complex, high-cost advanced evaluation framework that relies on Google Search. To enable quick and efficient evaluation, we instead use VeriScore to assess the LongFact dataset in this paper. VeriScore is an improved version of FActScore and SAFE, which optimizes the process by focusing on extracting and verifying "verifiable claims" and introducing inter-sentence context to improve extraction quality, thereby avoiding unnecessary revision steps. In this paper, we consistently use the powerful Qwen2.5-72B-Instruct to complete all steps of the VeriScore evaluation.

Factory. Factory[5] is a large-scale, human-verified, and challenging prompt set. We randomly select 250 samples from its "Hard" subset for evaluation. Similar to LongFact, we compute all three metrics using the VeriScore pipeline with a local Wiki knowledge base and the Qwen2.5-72B-Instruct model.

To ensure the reproducibility of our experiments and the transparency of the evaluation pipeline, we have compiled a detailed summary of the key parameter configurations used throughout the process. These configurations cover every step from fact decomposition to final verification. The specific parameter settings are detailed in Table 7.

Table 7: The key parameter configurations used in the evaluation pipeline.

Claim Extraction	Wiki Retrieval	Claim Verification
temperature=0.1 max_tokens=8192	top_k=10 chunk_size=300 chunk_overlap=20	temperature=0.1 max_tokens=8192

C.4.2 Metrics

This section provides formal definitions of the evaluation metrics used in our study to assess the long-form factuality of model responses. The reported values for Precision, Recall@K, and F1@K in our experiments are the average of each metric calculated for every individual response in the test set.

Precision. Precision measures the factual accuracy of a generated response. It is defined as the proportion of individual facts within the response that are verifiable as supported against an external knowledge source (e.g., a Wikipedia index). Let $S(y)$ be the number of supported facts in a response

[2] https://github.com/shmsw25/FActScore

[3] https://github.com/facebookresearch/HalluLens

[4] https://github.com/google-deepmind/long-form-factuality

[5] https://huggingface.co/datasets/facebook/FACTORY

y, and $N(y)$ be the number of not-supported facts. Precision for a single response is calculated as:

$$\text{Precision}(y) = \frac{S(y)}{S(y) + N(y)} \tag{15}$$

A higher precision indicates that the responses contain fewer factual errors on average.

Recall@K. Recall evaluates the comprehensiveness of a response. In open-domain long-text generation, defining the complete set of expected facts is infeasible. Following prior work [53], we adopt a parametric recall metric. Let K be a hyperparameter representing a user's desired number of supported facts for a high-quality response. Recall@K for a single response is calculated as:

$$\text{Recall@K}(y) = \min\left(\frac{S(y)}{K}, 1\right) \tag{16}$$

This metric measures whether a response provides an adequate amount of verifiable information, up to a specified limit K.

F1@K. To balance the trade-off between factual precision and informational comprehensiveness (Recall@K), we use their harmonic mean to compute the F1@K score for a single response:

$$\text{F1@K}(y) = \begin{cases} \frac{2 \cdot \text{Precision}(y) \cdot \text{Recall@K}(y)}{\text{Precision}(y) + \text{Recall@K}(y)}, & \text{if } S(y) > 0, \\ 0, & \text{if } S(y) = 0. \end{cases} \tag{17}$$

A response achieves a high F1@K score only by being both highly factual and sufficiently detailed.

WR. Win Rate (WR) is a comparative metric designed to evaluate the performance of a candidate model (B) relative to a baseline model (A). For each prompt in the test set, responses from both models are evaluated by a judge LLM (GPT-4.1 using Prompt 8 [29]), which receives two outputs and ranks them by which response would be preferred by humans. We note explicitly that this is a holistic preference protocol inherited from AlpacaEval: the judge is not instructed to extract atomic claims, verify them against external evidence, or prioritize factual correctness over fluency and completeness. WR is therefore not a claim-level factuality measure, and we do not use it as such. To control for position bias, each pairwise comparison is performed twice under reversed output ordering. In the first trial, Model A's response is provided as output1 and Model B's as output2; we record whether B is ranked higher than A, denoted as $\text{Win}^{(1)}$. In the second trial, the order is reversed: Model B's response is given as output1 and Model A's as output2, producing a result $\text{Win}^{(2)}$. The win indicator for the i-th instance is the average of both trials:

$$\text{Win}_i = \frac{\text{Win}^{(1)} + \text{Win}^{(2)}}{2} \tag{18}$$

The overall WR of model B over A on a test set of size N is defined as:

$$\text{WR}_{B \text{ vs } A} = \frac{\sum_{i=1}^{N} \text{Win}_i}{N} \tag{19}$$

A WR greater than 0.5 indicates that the candidate model (B) is preferred over the baseline (A). Unlike other metrics, WR provides a single aggregate measure of relative performance over the entire test set. Its role in this paper is diagnostic: because a precision-only policy can raise claim-level precision by shortening its answers, WR reveals whether such a policy is in fact perceived as worse overall. It complements, but never substitutes for, the claim-level metrics above.

C.5 Auxiliary Rewards

To ensure the overall quality of generated text beyond factuality and prevent potential degradation in readability or adherence to instructions, we introduce three auxiliary rewards.

General Reward. As we perform RL directly from the base model—bypassing SFT—a standard KL penalty is inapplicable. To prevent the policy from deviating into low-quality outputs, we employ a General Reward $R_g(\mathcal{A}_i)$ from Skywork-Reward-V2-Llama-3.2-1B [33] to incentivize responses that align with human preference.

Format Reward. We introduce a Format Reward R_f to enforce structured output in our thinking model. The output must encapsulate reasoning within <think></think> tags and the final answer within <answer></answer> tags:

$$R_f(o_i) = \begin{cases} 0, & \text{if } o_i \text{ has valid format,} \\ -1, & \text{otherwise.} \end{cases} \tag{20}$$

Length Penalty. To ensure the conciseness and information density of long-form factual responses, and to prevent the model from generating redundant content, we introduce a Length Penalty [57]. Let L_m denote the maximum allowed length threshold and L_c ($L_c < L_m$) be a predefined critical length value. The Length Penalty R_l is defined piecewise as follows:

$$R_l(\mathcal{A}_i) = \begin{cases} 0, & |\mathcal{A}_i| \le L_m - L_c, \\ \dfrac{(L_m - L_c) - |\mathcal{A}_i|}{L_c}, & L_m - L_c < |\mathcal{A}_i| \le L_m, \\ -1, & |\mathcal{A}_i| > L_m. \end{cases} \tag{21}$$

C.6 Reward Weight Selection

Factuality weights. Under GRPO the advantage is normalized within each group of rollouts for the same query, so what drives learning is not the absolute scale of a reward component but how much it varies within the group. The contribution of component k to the advantage ranking is thus proportional to $w_k \cdot \mathrm{std}(R_k)$, where $\mathrm{std}(R_k)$ is the within-group standard deviation, and equalizing the three contributions requires $w_k \propto 1/\mathrm{std}(R_k)$. Measuring these quantities before any RL training, on 2000 queries under the initial policy, gives $\mathrm{std}(R_{\text{truth}}) \approx 0.09 < \mathrm{std}(R_{\text{precision}}) \approx 0.13 < \mathrm{std}(R_{\text{recall}}) \approx 0.15$, whose normalized reciprocals $(0.26, 0.30, 0.44)$ give the constraint $\mu > \kappa \approx \gamma$, which we use to narrow the candidate set rather than to fix the weights. We trained three candidates satisfying this constraint, $(0.3, 0.3, 0.4)$, $(1/4, 1/4, 1/2)$ and $(0.2, 0.2, 0.6)$, plus two controls that violate it, $(1/3, 1/3, 1/3)$ and $(0.4, 0.2, 0.4)$, on Qwen2.5-14B from the base model under otherwise identical settings. Selection was made on a held-out set of 500 queries, disjoint from every evaluation benchmark, fixed before training, and never used to compute gradients; $(\kappa, \gamma, \mu) = (1/4, 1/4, 1/2)$ achieved the best validation F1 and is used unchanged for all reported experiments. The ablations in Table 2 further indicate that effectiveness does not hinge on precise tuning, since every configuration retaining all three components outperforms all baselines trained from the base model; the weights act as a control on the recall–precision trade-off rather than as a sensitive hyperparameter.

Auxiliary coefficients. The general reward uses a coefficient of 0.1. Its output is normalized to $[0, 1]$, so this places it on the same scale as the factuality rewards but at roughly one tenth of their magnitude, making it a soft regularizer that expresses a weak quality preference only among responses the factuality signals cannot distinguish. The format reward and length penalty use a coefficient of 1: both take values in $[-1, 0]$ and are zero for compliant outputs, so a penalty at the scale of the main rewards applies precisely when a constraint is violated, following the standard practice of encoding hard constraints as penalties rather than as competing objectives.

D Additional Experiments

In this section, we provide additional experimental details not covered in the main body, aiming to supplement and expand upon our findings. This content primarily includes supplementary analysis of our core experiments and other results not discussed in detail in the main text.

D.1 Verifier Training

Claim Extraction Model Training. To build an efficient and lightweight factual claim extraction model, we adopt the following training pipeline. First, we leverage the Claimify method proposed in [35], which has a significant advantage in generating structured factual claims. We use Qwen2.5-72B-Instruct as the data construction engine, utilizing a carefully designed prompt template to obtain verifiable claims. This process ultimately results in a large-scale, high-quality training dataset containing 10187 entries. Subsequently, we use Qwen2.5-14B-Instruct as the base model and perform supervised fine-tuning on this self-constructed dataset, yielding our final lightweight claim extraction

model. This model maintains high-precision extraction capabilities while significantly reducing computational overhead, allowing it to be seamlessly integrated into subsequent online reinforcement learning loops.

Checklist Verifier Training. To ensure the discriminative capability and generalizability of the Checklist Verifier, we train the model using a high-quality dataset of 10139 samples accumulated from previous research. First, we use DeepSeek-R1-0528 to generate entirely new and diverse responses for each training prompt. This step effectively enhances the model's robustness in judging texts of varying styles. Subsequently, we perform post-processing on these generated responses by removing any chain-of-thought content. This directs the model's focus toward factual verification rather than mimicking reasoning processes. Finally, we conduct supervised fine-tuning on the processed dataset using Qwen2.5-14B-Instruct as the base model, resulting in an efficient and lightweight verifier model.

We evaluate the SFT model on two self-constructed test sets: an English set with 81 questions and a Chinese set with 147 questions. The detailed evaluation results are presented in Table 8.

Table 8: Accuracy of the Checklist Verifier on the English (81Q) and Chinese (147Q) test sets. We select the epoch-2 checkpoint as the final model.

	English-81Q	**Chinese-147Q**
epoch1	69/81=0.852	128/147=0.871
epoch2	**73/81=0.901**	**129/147=0.878**
epoch3	70/81=0.864	129/147=0.878

Truthfulness Reward Model Training. The training of the Truthfulness Reward Model constitutes a crucial component of our framework. Specifically, after verifying all claims using Qwen2.5-72B-Instruct and the Wiki20250716 knowledge base, we obtain the verification results as summarized in Table 9, where "SUPPORT" and "REFUTE" represent positive and negative samples, respectively.

Table 9: Distribution of claim verification results based on Qwen2.5-72B-Instruct and Wiki20250716.

	ELI5			**LongFact-Gen**			**LongWiki-Gen**		
	SUPPORT	REFUTE	NOT ENOUGH INFO	SUPPORT	REFUTE	NOT ENOUGH INFO	SUPPORT	REFUTE	NOT ENOUGH INFO
7B	298822	14758	57376	149420	12854	46928	50260	13408	38990
14B	319512	9220	44304	168417	7757	36744	63920	9415	33760
32B	308659	7254	39929	163364	6903	33607	60035	9980	32779

Given the significant scarcity of negative samples compared to positive ones, we adopt the following sampling strategy: all negative samples are duplicated three times, while positive samples are down-sampled to achieve a positive-to-negative ratio of 2:1. Through this process and based on Prompt 6, we construct training datasets suitable for models of three different scales: 7B, 14B, and 32B. We then conduct SFT on pre-trained models of these three sizes to obtain reward models capable of assessing claim truthfulness based on the model's inherent knowledge boundaries.

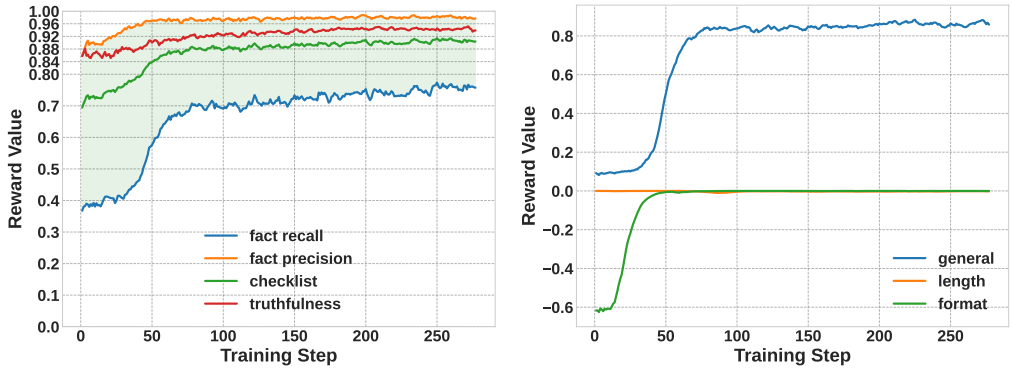
After model training is completed, we perform a systematic evaluation of their performance on independently constructed test sets. For the three model sizes—7B, 14B, and 32B—we construct separate test sets, each containing 3000 cases with a positive-to-negative sample ratio of 1:1. The test results of the models are detailed in Table 10. The table shows that the models after SFT exhibit significant improvements in both Accuracy and F1-score compared to the base model and the base few-shot model.

D.2 Main Results

In this section, we present a more detailed analysis of the training process and model performance through the reward curves in Fig. 4.

Table 10: Performance comparison of Truthfulness Reward Models at different scales on the test sets. The results show that models after SFT significantly outperform both the base model and the base few-shot model in terms of Accuracy and F1-score.

	Model	Accuracy	F1
7B (3000Q)	Base	0.7237	0.7684
	Base-few-shot	0.7813	0.7741
	SFT	**0.8187**	**0.8272**
14B (3000Q)	Base	0.6450	0.6321
	Base-few-shot	0.8096	0.8119
	SFT	**0.8347**	**0.8421**
32B (3000Q)	Base	0.7777	0.7752
	Base-few-shot	0.8200	0.8210
	SFT	**0.8334**	**0.8423**



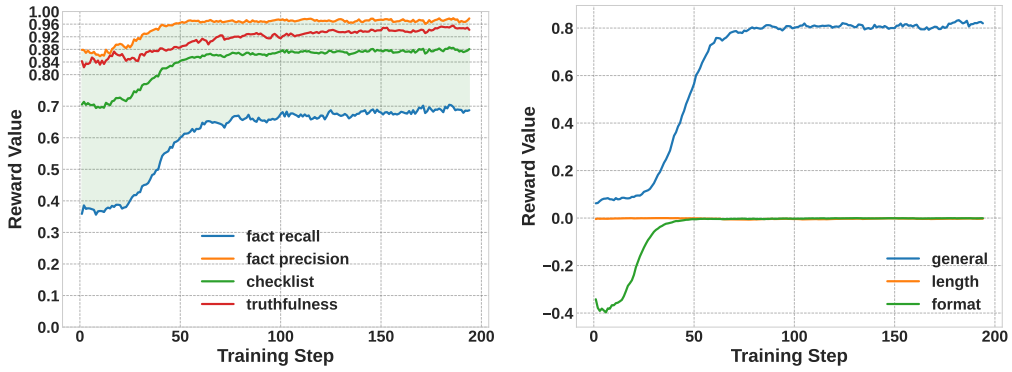
(a) Knowledge-Level Consistency Rewards.

(b) Auxiliary Rewards.

Figure 4: Training Dynamics of KLCF-zero on Qwen2.5-14B. (a) The core Knowledge-Level Consistency Rewards, all showing significant improvement. (b) The auxiliary rewards guiding response quality and structure.

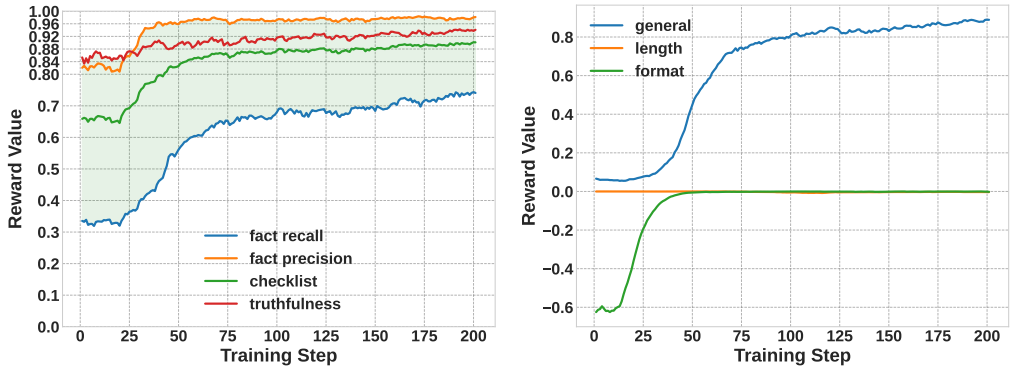
Knowledge-Level Consistency Rewards. All factual rewards show a significant upward trend, indicating that KLCF-zero training effectively optimizes the model's core objectives. Among them, the Fact Recall reward increases markedly from approximately 0.4 to around 0.75, demonstrating the greatest improvement. This confirms that the factual coverage (comprehensiveness) of the model's generated responses improves substantially, as the model learns to recall and output more facts within its knowledge boundaries. The Fact Precision reward rises further from a relatively high starting point (0.88) to near 0.98, indicating that the model maintains extremely high factual accuracy (correctness) while becoming more comprehensive. The Truthfulness reward remains at a very high level throughout (0.85 $\rightarrow$ 0.95), reflecting the base model's strong inherent truthfulness foundation, which our method further enhances.

Auxiliary Rewards. Changes in the auxiliary rewards reflect optimization of the model's overall quality. The General Reward (from the Skywork-Reward model) steadily increases from 0.1 to 0.8, indicating significant improvement in the overall quality, fluency, and human preference alignment of the generated responses. The Format Reward converges stably from an initial penalty of approximately -0.6 to 0, indicating that the model initially made formatting errors but quickly learned and fully mastered the required output structure (<think> ...</think><answer> ...</answer>), eventually incurring no further penalty. The Length Penalty fluctuates slightly near 0 throughout, demonstrating that the generation length remains effectively constrained within the predefined optimal range ($L_m - L_c = 850$ tokens), avoiding redundant or overly short responses.



(a) Knowledge-Level Consistency Rewards. (b) Auxiliary Rewards.

Figure 5: Training Dynamics of KLCF-zero on Qwen2.5-7B.



(a) Knowledge-Level Consistency Rewards. (b) Auxiliary Rewards.

Figure 6: Training Dynamics of KLCF-zero on Qwen2.5-32B.

Response Length. The total Response Length starts at around 100 tokens, gradually increases as training progresses, and eventually stabilizes around 700 tokens. The increase in answer length is primarily driven by the recall reward, which directly incentivizes the model to cover more facts from the checklist within the <answer></answer> section. Meanwhile, our designed length penalty effectively prevents answers from growing indefinitely, ensuring the conciseness and information density of the generated content.

Regarding the shorter CoT. All core factual rewards are computed solely based on the final answer and do not apply any optimization objective to the CoT itself. For general tasks like long-form QA (as opposed to complex mathematical reasoning), a longer preceding reasoning process does not necessarily lead to a better or more accurate final answer [50], and an ill-structured CoT can even increase the likelihood of hallucinations [8]. Furthermore, we observed that this phenomenon varies under different initial conditions (Base or SFT). In our experiments, when starting RL directly from the Base model, the model learned to use more concise reasoning chains to organize its thoughts during training, thereby allocating more expressive capacity to the answer section containing substantive facts. When RL is warm-started from an SFT model, the CoT length does not change significantly. We believe effective CoT patterns for factual tasks warrant further research.

D.3 Checklist Quality and Robustness

Because the factual checklist is a finite, protocol-conditioned approximation of $K_{\mathrm{vfe}}(q)$, the reliability of KLCF depends on how tolerant it is to an imperfect checklist. This section reports the construction quality of the checklist, controlled perturbation experiments, and the sensitivity of the resulting reward to the sampling protocol.

Construction quality. The checklist is built by the four-stage pipeline of Section 2.2.3: repeated base-model sampling, claim extraction, external verification against the Wiki20250716 index, and deduplication filtering. Only SUPPORT-labeled claims are retained, and the verifiers used in this pipeline are evaluated independently in Section D.1. We emphasize that the main experiments already operate under a heavily incomplete approximation: the 14B base model yields 38.34 supported claims per query at $\nu = 8$, of which strict filtering retains only 11.03 as checklist entries, roughly 29%. KLCF nonetheless attains the best F1 across all four benchmarks under this approximation, which is itself evidence that the method does not require an exhaustive checklist.

Controlled perturbations. To measure sensitivity directly, we perturb the checklist in two ways and retrain Qwen2.5-14B under the KLCF-zero setting. *Random deletion* removes a fraction of entries per query and simulates incompleteness. *Error injection* replaces a fraction of entries with irrelevant claims drawn from other queries and simulates verification errors. As a complementary test in the opposite direction, we also enlarge the checklist by raising the number of base-model samples from $\nu = 8$ to $\nu = 16$. Results on LongFact are given in Table 11.

Deletion and injection degrade performance in different ways, consistent with the role each signal plays. Removing entries mainly reduces recall while precision stays stable, since fewer facts are explicitly incentivized. Injecting false entries mainly reduces precision, since the recall reward now credits coverage of unsupported claims. In both cases the degradation is mild: under the two most severe settings, deleting half the entries or corrupting one fifth of them, F1@64 remains at 0.716 and 0.714, respectively, still well above 0.663, the best result achieved by any baseline trained from the base model. In the opposite direction, a 43% larger checklist improves F1@64 by only 1.1% and FActScore from 61.2% to 62.1%, indicating that $\nu = 8$ already lies on a performance plateau. Across the full range from deleting half the entries to doubling the sample count, F1@64 varies within 0.716–0.741.

Table 11: Robustness of KLCF to checklist incompleteness and noise, measured by LongFact F1@64 under the Qwen2.5-14B KLCF-zero setting. Error injection replaces entries with irrelevant claims from other queries, so the entry count is unchanged. For reference, the strongest baseline trained from the base model reaches F1@64 $= 0.663$ (Table 1).

Checklist setting	**Entries/query**	**F1@64**	Δ
Default ($\nu = 8$)	11.03	0.733	-
Random deletion, 30%	7.72	0.725	-1.1%
Random deletion, 50%	5.52	0.716	-2.3%
Error injection, 10%	11.03	0.726	-1.0%
Error injection, 20%	11.03	0.714	-2.6%
Enlarged ($\nu = 16$)	15.80	0.741	$+1.1\%$

Why optimizing on a subset generalizes. A further observation explains why an incomplete checklist still provides a useful optimization direction. During training, R_{recall} explicitly rewards covering only the roughly 11 checklist facts per query, yet the trained model reaches Recall@64 $= 0.776$ on the test set, which corresponds to at least 49.7 externally verified supported facts per response on average. The great majority of the recall gain therefore comes from correct knowledge that was never in the checklist. The behavior that R_{recall} encourages, namely expressing verified parametric knowledge more fully, generalizes to knowledge regions the checklist does not cover, because checklist facts and other correct knowledge share the same pre-training distribution. Optimizing on a checklist subset thus amounts to estimating the gradient from a representative sample of the knowledge space, which is sufficient to drive global improvement as long as the sample is aligned with the global objective. This is consistent with prior findings that RL fine-tuning primarily activates latent knowledge rather than injecting new information [25, 58].

Sensitivity to the sampling protocol. The checklist depends on three protocol parameters: the number of base-model samples ν, the few-shot context, and the sampling temperature. All three ultimately act through the same channel, namely how much of the base model's expressible knowledge ends up covered, which is exactly what the perturbation experiments vary. The effect of ν is measured directly above; deleting half the entries corresponds to an effective $\nu \approx 4$, so performance varies

by less than 2.5% over the range $\nu \in [4, 16]$. The few-shot context primarily controls output format rather than factual content, and high-temperature sampling already diversifies the elicited facts; a different context would shift which facts appear, which is equivalent to partial deletion. Lowering the temperature would drop low-probability knowledge, again equivalent to deletion from the high-temperature checklist. The robustness measured in Table 11 therefore covers all three parameters.

Sources of robustness. Two properties of the design account for this tolerance. First, the truthfulness reward acts as a global constraint: when R_{recall} encourages covering a corrupted entry, R_{truth} assigns the resulting claim a low score, and the two partially cancel in the advantage. Second, GRPO normalizes rewards within each group, which statistically dilutes the influence of individual noisy entries so long as most entries are correct. Dual-Fact Alignment is in this sense a systematic response to the unreliability of any single signal: the checklist supplies local high-precision supervision and the truthfulness reward model supplies a global constraint, and their complementarity keeps local errors in either from dominating training.

D.4 Efficiency Analysis

Table 12: Efficiency comparison of different factual reward calculation methods. All reported values are averaged over 50 samples. Our proposed reward (Checklist (recall + precision) + Truthfulness) achieves significant speedups in both serial and parallel modes. The "# Searches" column indicates the number of times the method called the external search engine API. Notably, our method requires no such calls and maintains low input & output token consumption.

	Reward	Time (s)	↑ (vs. FActScore)	↑ (vs. VeriScore)	# Searches	Input Tokens	Output Tokens
Serial	FActScore	198.06	1.00×	1.18×	51.74	55666.20	5701.56
	VeriScore	234.50	0.84×	1.00×	32.64	335909.78	3442.96
	Ours	57.29	**3.46**×	**4.09**×	0	1457.76	855.24
Parallel	FActScore	10.58	1.00×	1.03×	52.02	56677.50	5926.04
	VeriScore	10.90	0.97×	1.00×	32.56	334077.92	3407.28
	Ours	2.02	**5.24**×	**5.40**×	0	1452.62	851.46

Table 13: Pipeline-level cost comparison between KLCF and FActScore-based reward computation on Qwen2.5-14B. Offline preparation runs once on 4 × H100 and its outputs (factual checklists, truthfulness reward model, checklist verifier) are permanently reusable across RL runs, model scales, and tasks; online RL runs on 16 × H100. Rollout and update costs are identical for both methods, so the entire per-step gap originates in reward computation, where KLCF queries only local reward models.

	KLCF (ours)	**FActScore-based**
Offline preparation (one-time)	∼20 h	not required
Policy rollout (per step)	∼35 s	∼35 s
Reward computation (per step)	∼127 s	∼1056 s
Advantage estimation + update (per step)	∼50 s	∼50 s
Total per step	**∼212 s**	**∼1141 s**
External search API calls	✗	✓

This section provides additional implementation details for the efficiency analysis presented in Section 3.8, ensuring the fairness, reproducibility, and credibility of our results.

Data and Model Configuration. To guarantee a broad and unbiased evaluation, we randomly select 50 prompts from the training set of the Qwen2.5-14B model to form the evaluation dataset. To completely eliminate the influence of the model's training state on the length and structure of the generated content, we use an untrained DeepSeek-R1-Distill-Qwen-14B to generate responses for all prompts. These responses serve as the unified input for all reward calculation methods.

Our Reward Calculation Setup. For the calculation of our proposed rewards, the scale of the reward models and verifiers remains consistent with the main experiments. Specifically, the Checklist Verifier and the Truthfulness Reward Model are fine-tuned from Qwen2.5-14B-Instruct and Qwen2.5-14B, respectively, ensuring that the efficiency evaluation aligns with the actual application scenario.

Baseline Methods Calculation Setup. To ensure a fair comparison, we standardize the internal components of the baseline methods, FActScore and VeriScore. We replace their original claim extraction and claim verification modules with the same-sized Qwen2.5-14B-Instruct. This step eliminates the potential impact of model scale differences on computational efficiency, focusing the comparison solely on the overhead introduced by the algorithmic designs.

Deployment and Computational Environment. We deploy all models using the vLLM inference framework on two H100 GPUs. When evaluating the parallel computing performance of all methods, we set the concurrency to 100 to effectively simulate the high-throughput reward calculation scenario common in large-scale reinforcement learning.

As shown in Table 12 and Table 13, our KLCF reward design substantially reduces computational overhead and eliminates critical deployment bottlenecks.

Reward computation is the dominant bottleneck in online RL training. In every training iteration, policy rollout takes $\sim$35 seconds, while reward calculation dominates the total step time. Our KLCF reward requires only $\sim$127 seconds, compared to $\sim$1056 seconds for FActScore. This means our reward computation is only **12% of the cost of FActScore**, leading to **5.4$\times$ faster per-iteration training** in practice.

One-time offline investment yields long-term, scalable gains. Although offline preparation takes $\sim$20 hours (building checklists and training reward models), the resulting assets (checklists and reward models) are reusable across unlimited RL runs, model scales, and even tasks. To quantify: for a 100-step RL run, KLCF totals "100 $\times$ 212 s = 5.89 hours" of online time, whereas FActScore totals "100 $\times$ 1141 s = 31.69 hours". **A single training run already saves $\sim$25.8 hours.** For large-scale training or hyperparameter sweeps, the savings rapidly dwarf the one-time offline cost.

Our design avoids real-world deployment bottlenecks. FActScore-based methods rely on external search APIs, which suffer from rate limits, network latency, and instability—often becoming the slowest component in distributed RL pipelines. Moreover, in mixed-task training (e.g., combining factual and mathematical reasoning), the entire pipeline must wait for the slowest reward signal. Our method uses only lightweight, locally deployed reward models, eliminating external API dependencies and ensuring stable, high-throughput reinforcement learning at scale.

In summary, while we acknowledge the one-time offline preparation cost, our method shifts the primary overhead to a reusable investment and eliminates the critical online bottleneck, achieving substantially higher overall efficiency—especially in large-scale and mixed-task RL settings.

D.5 Truthfulness Reward Analysis

This section compares the standard Truthfulness Reward (Eq. (8)) with its variant (Eq. (9)) under the RL-Zero setting on Qwen2.5-14B. The variant was motivated by noise reduction: claims already covered by the offline-verified checklist have externally confirmed factuality, so rather than scoring them with the reward model, the variant excludes them from the average and scores only the uncovered subset, returning full credit when nothing is uncovered. In practice it is consistently slightly weaker (Table 14), and the gap originates in the one component the standard reward does not have: the coverage decision.

That decision runs opposite to the direction our verifier was trained for. The verifier judges whether a response expresses a given checklist fact, which is what the checklist reward requires, whereas the variant must judge whether the checklist already contains a given claim from the response. We reuse the same verifier without retraining, so its input distribution shifts from concise, curated checklist entries to arbitrary model-generated claims, degrading the decision. When a covered claim is wrongly judged uncovered, the variant both forfeits the high-confidence prior it was designed to exploit and penalizes the policy for expressing content that was in fact externally verified.

Enlarging the checklist localizes the gap to this component. Raising the number of base-model samples from $\nu = 8$ to $\nu = 16$ expands the checklist from 11.03 to 15.80 entries per query and

Table 14: Comparison between the standard truthfulness reward and its variant, both with the default checklist ($\nu = 8$). Experiments were conducted by performing RL from the base Qwen2.5-14B model. WR is a holistic preference metric, as noted in Table 1.

Reward	FActScore	LongWiki			LongFact				Factory			
	FS	R@32	Prec	F1@32	R@64	Prec	F1@64	WR	R@64	Prec	F1@64	WR
Base (10-shot)	46.8%	0.661	0.435	0.511	0.614	0.685	0.637	-	0.239	0.302	0.262	-
R_{truth}	**61.2%**	**0.681**	**0.552**	**0.568**	**0.776**	**0.704**	**0.733**	**94.6%**	**0.296**	**0.350**	**0.309**	**65.5%**
$R_{\text{truth}}^{\text{variant}}$	60.7%	0.643	0.548	0.545	0.751	0.691	0.714	91.0%	0.284	0.350	0.299	56.2%

reduces the LongFact F1@64 gap from 1.9 points (0.733 versus 0.714) to 0.6 points (0.741 versus 0.735; the standard-version figure is the enlarged-checklist row of Table 11). The variant's larger improvement is concentrated in recall, consistent with the coverage decision having been the binding constraint, since misjudged coverage suppresses exactly the expression of verified content. We therefore use the standard reward in all main experiments: by scoring every claim uniformly it requires no coverage decision at all, which makes it both cheaper and free of this failure mode. Retraining the verifier for the reverse direction would be the natural way to make the variant competitive, and we leave this to future work.

D.6 Case Study

In this section, we conduct a qualitative study on the performance of the KLCF framework on specific samples. Specifically, we compare Qwen2.5-14B-Instruct and its KLCF-trained version on an example from the LongFact test set. The results are shown in Table 15.

The analysis shows that the output of the model after KLCF training is significantly longer and more detailed. This directly reflects a substantial improvement in **factual recall**, indicating that the model more actively and completely utilizes the information in its pre-trained knowledge base, rather than adopting a conservative strategy. This improvement primarily stems from our proposed Factual Checklist Reward mechanism, which incentivizes the model to cover a broader range of known facts during generation, thereby effectively mitigating over-conservatism.

More importantly, while the output content is expanded, its **factual precision** is also improved. The output of the model without KLCF training contains some vague descriptions and information gaps, whereas the output of the KLCF model not only includes more factual details but also describes them more accurately and specifically, without introducing observable new errors. This validates the effectiveness of the Dual-Fact Alignment mechanism: the Truthfulness Reward, based on the model's internal knowledge for self-assessment, guides the model to generate claims with higher confidence, thus effectively reducing the production of inaccurate or fabricated content.

In summary, this case qualitatively demonstrates that KLCF enables the model to achieve a better balance between factual recall and precision.

Table 15: Responses to the query "*Who is lawyer Amal Clooney?*" from the Qwen2.5-14B-Instruct model and its KLCF-enhanced counterpart. Text highlighted in **red bold** across both responses indicates factually inaccurate claims. The KLCF-generated response is notably longer and more comprehensive, encompassing a greater number of verifiable factual details while containing fewer overall factual errors.

Qwen2.5-14B-Instruct Response:

Amal Clooney is a renowned international human rights lawyer and advocate based in London and New York. She is known for her work on high-profile cases involving human rights violations, freedom of expression, and other significant legal issues around the world. Amal has represented individuals and organizations in various countries, including Lebanon, Egypt, and Azerbaijan, often focusing on cases that involve press freedom and the rights of journalists.

She was born on April 18, 1978, in Beirut, Lebanon, and **grew up there before moving to England at age 13**. Amal attended **St. Paul's Girls' School** and later studied law at **St Edmund Hall**, Oxford University, and **Middle Temple**, where she was called to the bar in 2002. She went on to complete an LLM at **Columbia Law School in New York City**.

In addition to her legal work, Amal Clooney is also recognized for her philanthropic efforts and advocacy for various causes. She is married to the actor George Clooney, with whom she shares two children. Together, they have been involved in several charitable initiatives and campaigns aimed at promoting human rights and social justice.

+ KLCF Response:

Amal Clooney is a renowned international human rights lawyer and advocate known for her work on high-profile cases and her commitment to humanitarian issues around the world. Born on February 3, 1978, in Beirut, Lebanon, she has gained significant public attention not only due to her professional achievements but also because of her marriage to the actor George Clooney, though her career and legal work stand out as her primary contributions.

Key Aspects of Amal Clooney's Career

1. **Education and Early Career**: Amal Clooney earned her undergraduate law degree in Jurisprudence at St Hugh's College, Oxford, and later obtained an LLM from New York University School of Law. Her academic background laid a strong foundation for her legal career, focusing on international law and human rights advocacy.

2. **Legal Expertise**: She specializes in international law, particularly in areas such as freedom of expression, press freedom, and human rights violations. She has represented clients in cases before the International Criminal Court (ICC), European Court of Human Rights, and other international tribunals.

3. **High-Profile Cases**: Clooney has been involved in several notable cases that have garnered global media attention. These include defending journalists and activists against charges that could be seen as politically motivated or unjust, such as the case of the Al Jazeera journalists detained in Egypt, and advocating for victims of war crimes and human rights abuses.

4. **Humanitarian Work**: Beyond her legal practice, she works with various organizations to promote human rights and supports initiatives aimed at protecting journalists and whistleblowers. She has been vocal about issues related to refugees and has worked on campaigns to raise awareness and improve conditions for those seeking asylum.

5. **Public Advocacy and Speeches**: She often speaks at international forums, conferences, and universities...

6. **Philanthropy and Social Causes**: Clooney is associated with multiple charities and foundations that focus on education...

7. **Professional Networks and Collaborations**: She collaborates with international legal bodies, non-governmental organizations...

8. **Personal Life and Public Image**: While her personal life, including her marriage to George Clooney...

Impact and Legacy

Amal Clooney's impact extends beyond individual cases; she contributes to shaping legal strategies and international norms regarding human rights and justice. Her work helps set precedents that can influence how governments and international bodies handle human rights issues and prosecute crimes against humanity.

Her approach to law combines rigorous legal analysis with a strategic use of media and public opinion to achieve justice for her clients and to push for broader systemic changes. This dual approach—combining traditional legal advocacy with modern communication strategies—has made her one of the most influential figures in international human rights law.

Challenges and Criticisms

Despite her accolades, Clooney faces criticism from some quarters regarding the effectiveness of her methods, the balance between her public persona and substantive legal outcomes, and occasional controversies surrounding the high-profile nature of her cases versus more systemic issues faced by less visible groups.

Overall, Amal Clooney is recognized for her dedication to advancing human rights and justice on a global scale, using her skills and platform to address complex legal and ethical challenges facing our interconnected world. Her work continues to inspire and influence legal practitioners and human rights advocates worldwide.

E Prompts

This section provides all prompts employed in this study. The **red bold** portions within the prompts indicate placeholders to be filled.

E.1 Data Construction

Prompt 1: Base Model Generation

Please answer the following questions in a given format.
[Question]: query example 1
[Answer]: response example 1

[Question]: query example 2
[Answer]: response example 2
...
[Question]: query example 10
[Answer]: response example 10

[Question]: **{query}**
[Answer]:

Prompt 2: Claim Extraction

Below you will receive a piece of text. Your task is:

1. Determine whether the text contains verifiable objective claims.
2. If verifiable objective claims exist in the text, you must extract these claims from the answer (regardless of whether these claims are true).
3. If the text does not contain any verifiable objective claims, return "no verifiable objective claims".

Response format:

* Claim 1
* Claim 2
...
(or "no verifiable objective claims")

The claims you extract must adhere to the following 3 principles:

1. Objectively verifiable: The claim must describe an objectively verifiable fact, not a subjective judgment, evaluation, or opinion.
2. Indivisible: The objective fact described by the claim cannot be further broken down.
3. Explicit meaning: Each claim must be a complete, self-contained sentence with all coreferences resolved. There should be no nouns or pronouns with unclear meaning.

Please strictly follow the above rules to complete the following task:
[Text]: **{response}**
[Verifiable objective claims]:

Prompt 3: Claim Verification

You will be provided with a [CLAIM] and several pieces of reference [EVIDENCE]. Your task is to determine the relationship between the facts in the [CLAIM] and the [EVIDENCE].

You need to analyze each piece of [EVIDENCE] in relation to the given [CLAIM] and then provide an overall conclusion.

- **SUPPORT**: The facts in the [CLAIM] appear in the [EVIDENCE] and are consistent with the descriptions. Or, the facts in the [CLAIM] can be correctly inferred from the information in the [EVIDENCE].
- **REFUTE**: The facts in the [CLAIM] contradict the information in the [EVIDENCE]. Or, the facts in the [CLAIM] can be inferred to be false based on the information in the [EVIDENCE].
- **NOT ENOUGH INFO**: The facts in the [CLAIM] are neither supported by nor contradicted by the information in the [EVIDENCE], or a definitive conclusion cannot be drawn from the [EVIDENCE].

Your response must be a dictionary with a single key-value pair, where the key is "conclusion" and the value is the overall conclusion (**SUPPORT**, **REFUTE**, or **NOT ENOUGH INFO**). You must strictly follow the format below, returning a dictionary in JSON format. Do not return any other content.

[RESPONSE FORMAT]:
```json
{
    "conclusion": "The overall conclusion, return one of **SUPPORT**, **REFUTE**, or **NOT ENOUGH INFO**"
}
```

Now, complete the following task.
[CLAIM]: **{claim}**

[EVIDENCE]:
{evidence}

Your decision:

Prompt 4: Claim Prioritization

You are given a user query and a list of candidate claims.

Your task:
- Retain only the claims that are essential and highly relevant to the query.
- Eliminate duplicates. If two claims capture slightly different aspects, preserve both.
- Each claim must convey exactly one idea, expressed as a clear, explicit, and self-contained sentence.
- If conflicting claims exist, present them in the form: "Some evidence suggests [...], while others indicate [...]".
- Output format:
\boxed{
    key claim 1
    key claim 2
    key claim 3
    ...
  }
- Provide output strictly within \boxed{}.

Output Format:
\boxed{
    Final Key Claim 1
    Final Key Claim 2
    ...
  }

Input:
- Query: **{query}**
- Candidate Claims:
{claims}

Output:

E.2 Training

Prompt 5: RL-Zero

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within <think></think> and <answer></answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>. User: **{prompt}** Assistant: <think>

Prompt 6: Truthfulness Reward Model Training

You need to evaluate the factual correctness of the following claim and output either "True" (the claim is entirely factually correct) or "False" (the claim contains errors or goes beyond your knowledge).

Claim: **{claim}**
Factual correctness:

Prompt 7: Checklist Verification

You will receive a **Question**, a **Reply** and a **Fact List**. Your task is to check, one by one, how each fact in the list is covered in the reply and decide whether the fact is **Consistent**, **Contradictory**, or **Missing**.

Input

* [Question]: a factual question.
* [Reply]: the text answer that needs to be evaluated.
* [Fact List]: a list of factual points to verify.

Output

Return a list in **strict JSON format**. For every fact in the list, output a dictionary containing:

* "analysis": a brief analysis of how the reply aligns with this fact.
* "conclusion": one of "Consistent", "Contradictory", or "Missing", indicating how the fact is covered in the reply.

* **"Consistent"**: the core information of the fact appears in the reply and matches the description.
* **"Contradictory"**: some information in the fact conflicts with information in the reply; from this fact you can infer that part of the reply is incorrect.
* **"Missing"**: the fact is neither fully implied nor contradicted by the reply; the reply does not mention the fact or omits some core information.

Now, using the **Question**, **Reply**, and **Fact List** given below, analyze each fact and output the results strictly in the required format:

[Question]: **{query}**
[Reply]: **{response}**
[Fact List]:
{guidelines}

[Output]:

E.3 Evaluation

Prompt 8: Calculating Win Rate

<|im_start|>system
You are a helpful assistant, that ranks models by the quality of their answers.
<|im_end|>
<|im_start|>user
I want you to create a leaderboard of different of large-language models. To do so, I will give you the instructions (prompts) given to the models, and the responses of two models. Please rank the models based on which responses would be preferred by humans. All inputs and outputs should be python dictionaries.

Here is the prompt:
{
"instruction": """**{instruction}**"""
}

Here are the outputs of the models:
[
{
"model": "model_1",
"answer": """**{output_1}**"""
},
{
"model": "model_2",
"answer": """**{output_2}**"""
}
]

Now please rank the models by the quality of their answers, so that the model with rank 1 has the best output. Then return a list of the model names and ranks, i.e., produce the following output:
[
{'model': <model-name>, 'rank': <model-rank>},
{'model': <model-name>, 'rank': <model-rank>}
]

Your response must be a valid Python dictionary and should contain nothing else because we will directly execute it in Python. Please provide the ranking that the majority of humans would give.
<|im_end|>

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The primary claims presented in the abstract and introduction accurately represent our paper's contributions and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations in Section 4.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: This paper does not contain formal theoretical results or proofs.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide complete reproduction details, including the source code (linked in the abstract), the data construction pipeline (Appendix C.1), training hyperparameters (Appendix C.3), and evaluation configurations (Appendix C.4).

   Guidelines:

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

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

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

Answer: [Yes]

Justification: Our code and trained models are publicly available via the links provided in the abstract. The data construction pipeline and all training and evaluation configurations are documented in Appendices C.1, C.3, and C.4.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: The main experimental settings are given in the Experiments section, and full details, including data splits, hyperparameters, optimizer settings, and evaluation configurations, are provided in Appendix C.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [No]

Justification: Due to computational constraints, we do not report error bars or statistical significance tests.

Guidelines:

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

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Compute resources (worker type, memory, execution time) are reported in Appendices C.3, C.4, and D.4.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Our work complies with the NeurIPS Code of Ethics; no human subjects, sensitive data, or ethical concerns are involved.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss both positive and negative societal impacts in Appendix A.2.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: Our models and data do not pose a high risk for misuse beyond general-purpose LLMs.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All existing assets (code, data, models) are properly cited with source attribution.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

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

13. **New assets**

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

    Answer: [Yes]

    Justification: Our new assets, including the source code and trained model checkpoints, are released with documentation via the links provided in the abstract.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Our work does not involve crowdsourcing or human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: No human subjects were involved in this work.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: We discuss LLM usage in Appendix A.1. Specifically, GPT-5 was used only as a writing aid for proofreading and polishing select passages, without contributing to core methodology, experiments, or conclusions.

    Guidelines:

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