TRAIN FOR TRUTH, KEEP THE SKILLS: BINARY RETRIEVAL-AUGMENTED REWARD MITIGATES HALLUCINATIONS

Anonymous authors
Paper under double-blind review

Revision Note: We highlighted our major changes with color.

ABSTRACT

Language models often generate factually incorrect information unsupported by their training data, a phenomenon known as extrinsic hallucination. Existing mitigation approaches often degrade performance on open-ended generation and downstream tasks, limiting their practical utility. We propose an online reinforcement learning method using a novel binary retrieval-augmented reward (RAR) to address this tradeoff. Unlike continuous reward schemes, our approach assigns a reward of one only when the model's output is entirely factually correct, and zero otherwise. We evaluate our method on Qwen3 reasoning models across diverse tasks. For open-ended generation, binary RAR achieves a 39.3% reduction in hallucination rates, substantially outperforming both supervised training and continuous-reward RL baselines. In short-form question answering, the model learns calibrated abstention, strategically outputting "I don't know" when faced with insufficient parametric knowledge. This yields 44.4% and 21.7% fewer incorrect answers on POPQA and GPQA, respectively. Crucially, these factuality gains come without performance degradation on instruction following, math, or code, whereas continuous-reward RL, despite improving factuality, induces quality regressions.

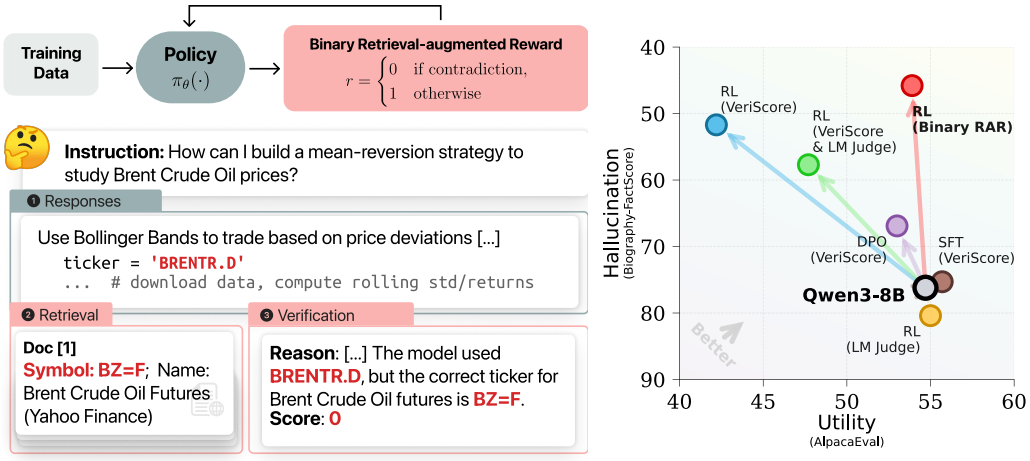

Figure 1: Overview of Binary Retrieval-Augmented Reward (Binary RAR). Left: Reinforcement learning with Binary RAR assigns a binary reward based on retrieval-verified factual correctness. Right: Binary RAR achieves the best hallucination–utility tradeoff among all post-training baselines. (Revision Note: we replace factual precision with hallucination rate.)

1 INTRODUCTION

Large language models (LMs) have transformed how people seek and process information, demonstrating remarkable capabilities in knowledge memorization and problem-solving (Chatterji et al., 2025). However, their widespread adoption is hindered by a critical reliability issue: extrinsic hallucination, where models generate seemingly plausible but factually incorrect information (Kalai et al., 2025; Li et al., 2024a). This problem has become increasingly concerning as recent state-of-the-art reasoning models exhibit higher rates of hallucination (Yao et al., 2025; Song et al., 2025).

Simply scaling up pre-training cannot resolve hallucination since pre-training optimizes next-token likelihood without enforcing factual correctness in generation (Kalai et al., 2025; Wen et al., 2025). Recent post-training efforts have explored several directions: supervised fine-tuning on carefully curated responses that consider the model's ability and express uncertainty when appropriate (Newman et al., 2025; Zhang et al., 2024), direct preference optimization (DPO) with factuality-focused preference pairs (Tian et al., 2024; Lin et al., 2024; Gu et al., 2025), and reinforcement learning (RL) with continuous factuality rewards (Liang et al., 2024; Chen et al., 2025). However, these approaches face a critical challenge: reducing hallucination often comes at the cost of overall utility. Models may generate less informative responses (Su et al., 2025; Wu et al., 2025), abstain excessively regardless of question difficulty (Cheng et al., 2024; Brahman et al., 2024), or degrade in general capabilities like instruction following (Lin et al., 2024). We target *continual post-training on fully trained models to mitigate hallucinations without degrading overall utility across varied tasks*, including instruction following, knowledge retention, reasoning, and coding.

In this paper, we address this hallucination-utility tradeoff through a novel approach: *online* RL with binary retrieval-augmented rewards (RAR; Figure 1 left). Unlike prior works using continuous factuality scores that can be vulnerable to reward hacking, we propose a simple binary signal: $r \in \{0, 1\}$ with $r = 0$ if any information in the output contradicts the retrieved documents, and $r = 1$ otherwise. To compute RAR, we retrieve candidate evidence from the web and evaluate the factual correctness of an LM's response in the rollout based on these documents, identifying conflicts rather than verifying based on a ground-truth answer. This design choice is inspired by successful applications of binary rewards in mathematical reasoning and coding tasks (Lambert et al., 2025; Shao et al., 2024). Our approach offers several key advantages. First, the binary reward structure inherently resists reward hacking by avoiding partial credit for stylistic changes that may mislead continuous scoring functions. Second, our single unified reward applies to both long-form generation and short-form question answering. Third, the framework naturally encourages appropriate abstention through RL's downweighting of incorrect answers, thereby upweighting abstention behavior inherited from the fully trained base model.

We train Qwen3 (Qwen-Team, 2025) reasoning models (4B and 8B) with our Binary RAR method and evaluate them on four hallucination benchmarks and ten general capability benchmarks, showing that Binary RAR effectively addresses the hallucination–utility tradeoff. As shown in Figure 1 right, in long-form generation, we reduce hallucination rates from 76.2 to 45.8, substantially outperforming DPO (66.9) and concurrent RL work with continuous VeriScore rewards (51.7; proposed by Chen et al. 2025). Crucially, we achieve this while maintaining general capabilities: ALPACAEVAL (Dubois et al., 2024) score remains largely stable (-1.4%), whereas continuous reward baselines show significant degradation (-22.8%). For short-form question answering, where Qwen3-8B models rarely abstain even when prompted to do so, our RL method reduces the hallucination rate from 60.6 to 27.6 while preserving accuracy when the model is asked to make its best guess. Similar patterns hold across model scales: on Qwen3-4B, binary RAR achieves 43.0% relative hallucination reduction in long-form generation, surpassing VeriScore at 29.1%. These results indicate that optimizing a binary, retrieval-verified signal yields larger hallucination reduction with fewer side effects on general capabilities compared to continuous factuality rewards.

Through detailed analysis, we find that models trained with Binary RAR retain informativeness while eliminating incorrect content selectively. In long-form generation, they maintain nearly the same number of correct claims but substantially reduce false ones, indicating improved precision rather than loss of detail. In short-form question answering, the models mostly retain their accuracy while largely reducing incorrect answers and increasing abstention, showing more controlled and calibrated response behavior. Our case studies further reveal that continuous reward formulations are vulnerable to stylistic biases and noise from retrieval or verification, whereas Binary RAR remains

robust to these factors. Overall, these results establish online RL with Binary RAR as a stable and effective approach to enhance factual reliability without compromising general capability.

2 Related Work

Measuring hallucinations in LM outputs Despite their impressive capabilities across diverse tasks, LMs are prone to hallucination, producing incorrect statements with unwarranted confidence (Mallen et al., 2023). The most widely adopted taxonomy distinguishes between two primary types of hallucination based on their relationship to provided prompts (Ji et al., 2023a; Huang et al., 2025; Bang et al., 2025). *Intrinsic hallucination* is defined as output that is inconsistent with the user's prompt or the provided input context. In this paper, we focus on *extrinsic hallucination*, which refers to generated output that cannot be verified from the training data. Measuring extrinsic hallucinations in long-form generation is particularly challenging due to its open-ended nature (Qi et al., 2025). Several distinct approaches have been proposed to automatically identify hallucinated content, including NLI-based methods (Gao et al., 2023; Min et al., 2023; Song et al., 2024), QA-based methods (Tian et al., 2024), uncertainty estimation (Farquhar et al., 2024; Orgad et al., 2025), and LLM-as-a-Judge (Li et al., 2024b). Following previous work, we adopt the approach of verifying atomic claims in the output as our evaluation method for long-form generation, which was first proposed in (Min et al., 2023). Specifically, we decompose a response into atomic, verifiable claims and then check each claim against related documents.

Reducing hallucination via post-training Many prior works explore mitigation methods at inference time, such as retrieval-augmented generation (Asai et al., 2024), prompting techniques (Ji et al., 2023b), and decoding algorithms (Chuang et al., 2024). In this work, we study how to fine-tune models during post-training to mitigate extrinsic hallucination. Supervised fine-tuning (SFT) can improve factuality by avoiding training on knowledge that the model has not already assimilated during pre-training, as fine-tuning on unfamiliar knowledge can increase the propensity for hallucination (Newman et al., 2025; Zhang et al., 2024). Similarly, Direct Preference Optimization (DPO) trains the model to prefer more factual responses over less factual ones (Tian et al., 2024; Lin et al., 2024). This is often achieved by generating response pairs where preferences are determined by continuous factuality assessment scores. Concurrent with this work, Chen et al. (2025) combine offline learning (SFT, DPO) with online RL to enhance base LMs' factuality using a continuous factuality signal (i.e., VeriScore). However, prior efforts largely emphasize factuality gains while offering limited assessment of impacts on other LM capabilities. In addition, several prior studies apply on-policy RL only to short-form question answering, where rewards are easier to define and do not extend naturally to long-form generation (Xu et al., 2024). We address this gap with an on-policy RL method that employs a *search-augmented binary reward*, improving the factuality of *fully trained* LMs *without* degrading general capabilities.

3 RL with Binary Retrieval-Augmented Reward

Our goal is to reduce hallucination while preserving the general capabilities of a fully trained LM. Prior approaches that leverage SFT or DPO to improve factuality are typically *offline*: they collect datasets once from human or previous model outputs instead of sampling new responses from the current model during training. We instead adopt *online RL*, computing rewards on the model's *own rollouts*, and introduce a novel *binary retrieval-augmented reward* (**Binary RAR**; Figure 1) that focuses on determining whether the entire response contains errors, with KL regularization to control drift.

This section presents the training objective and algorithmic setup (§3.1), defines and motivates the binary reward with retrieval and verification (§3.2), and describes the dataset curation (§3.3).

3.1 Preliminaries and Training Objective

The application of RL to LMs frames the training process as an optimization problem. Given a prompt x, an LM π_θ generates a response y according to a policy $\pi_\theta(y \mid x)$. The goal is to train the policy to maximize a reward function $r(x, y)$, which assigns a scalar score to the generated response. To prevent the fine-tuned model from deviating excessively from its original capabilities,

the optimization is typically constrained by a Kullback–Leibler (KL) divergence term against a reference model π_{ref}. The objective is formally expressed as:

$$\max_{\pi_\theta} \quad \mathbb{E}_{\substack{x\sim\mathcal{D} \\ y\sim\pi_\theta(\cdot|x)}} \Big[r(x,y) - \beta\, \mathbb{D}_{\text{KL}}\big(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\text{ref}}(\cdot \mid x)\big) \Big] \tag{1}$$

where $\mathcal{D}$ is the prompt dataset and β controls the strength of the KL penalty.

Several algorithms exist to optimize this objective. Among them, Group Relative Policy Optimization (GRPO; Shao et al. 2024) has become a popular choice for LM post-training due to its stability and computational efficiency (DeepSeek-AI et al., 2025). GRPO removes the critic model, which is typically as large as the policy model, and estimates the baseline from group scores instead. Specifically, for each prompt x, GRPO samples a group of outputs $y_1, ..., y_n$ from the old policy π_{old} and optimizes the policy model π_θ by maximizing:

$$\max_{\pi_\theta} \mathbb{E}_{\substack{x\sim\mathcal{D} \\ \{y_i\}_{i=1}^{n}\sim\pi_{\text{old}}(\cdot|x)}} \Bigg[\frac{1}{n}\sum_{i=1}^{n} \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \min\Bigg(\frac{\pi_\theta(y_i^t \mid y_i^{<t}, x)}{\pi_{\text{old}}(y_i^t \mid y_i^{<t}, x)} A_i,\ \text{clip}\Bigg(\frac{\pi_\theta(y_i^t \mid y_i^{<t}, x)}{\pi_{\text{old}}(y_i^t \mid y_i^{<t}, x)}, 1-\epsilon,\ 1+\epsilon \Bigg) A_i \Bigg) - \beta\, \mathbb{D}_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}}) \Bigg], \tag{2}$$

where ϵ and β are hyperparameters, and the advantage A_i and KL regularization $\mathbb{D}_{\text{KL}}$ are defined as:

$$A_i = \frac{r(x,y_i) - \text{mean}[\, r(x,y_1), ..., r(x,y_n)\,]}{\text{std}[\, r(x,y_1), ..., r(x,y_n)\,]} \tag{3}$$

$$\mathbb{D}_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}}) = \frac{\pi_{\text{ref}}(y_i \mid x)}{\pi_\theta(y_i \mid x)} - \log \frac{\pi_{\text{ref}}(y_i \mid x)}{\pi_\theta(y_i \mid x)} - 1 \tag{4}$$

We adopt GRPO as the default RL algorithm for our experiments.

3.2 Binary Retrieval-Augmented Reward

Overview. Our reward design targets hallucination reduction in both long-form and short-form generation. For long-form generation, we expect models to produce responses with minimal factual errors while maintaining high quality (e.g., as measured by an automatic LM judge). For short-form tasks, we expect models to acknowledge "I do not know" when they lack knowledge and to provide correct answers when possible. Overall, our goal is to downweight any response containing incorrect information while preserving correct or abstaining responses. We assign low scores to incorrect outputs and use an appropriate KL coefficient to retain the probability of correct answers from the base model. This corresponds to the reward and KL terms in Equation 1.

Pipeline. We define the factual correctness of an instruction–response pair (x, y) as the consistency between the generated content and reliable sources. A pair is considered correct if all information in y is supported by evidence. We introduce a binary retrieval-augmented reward $r(x, y) \in \{0, 1\}$ as follows. We use this binary RAR as a proxy for true factual correctness in the RL training (Figure 1, left).

- **Retrieval.** A datastore $\mathcal{DS} = \{d_i\}_{i=1}^{M}$ consists of reliable documents that are preprocessed, chunked, and indexed by a retriever R. To verify factuality, we retrieve the top k relevant documents for each (x, y) pair based on similarity $R(y, d)$, denoted as $C(x, y)$. These documents serve as evidence for verification.
- **Verification.** To check correctness, an LM verifier takes $(x, y, C(x, y))$ as input and determines whether contradictions exist between the response and retrieved documents. The verifier focuses solely on contradictions, given the context of x. Formally,

$$r(x,y) = \begin{cases} 1 & \text{if no contradictions are found between } (x,y) \text{ and } C(x,y), \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

We then optimize the KL-constrained RL objective (Equation 2) with this binary retrieval-augmented reward. This approach avoids the complexity of continuous reward design and provides a cleaner, less noisy training signal. Prompting details are given in Appendix D.

3.3 Efficiency Considerations for Training Scalability

Retrieval and Pre-caching Strategy. Both retrieval and verification are computationally intensive, and computing reward $r(x, y)$ can easily become the bottleneck of RL training. To improve efficiency, we adopt a pre-caching strategy. During dataset preparation, we pre-cache a set of relevant documents $\mathcal{DS}_{\text{cache}}(x)$ for each prompt x in the training set $\mathcal{D}$. At training time, we retrieve $C(x, y)$ from this cached subset rather than from the full datastore $\mathcal{DS}$. To build $\mathcal{DS}_{\text{cache}}(x)$, we query the Google Search API using the ground-truth response to retrieve up to 10 potentially relevant web pages, which we crawl and parse using a rule-based Python pipeline. Instances with fewer than three retrieved documents are discarded, as sparse evidence is often insufficient for reliable verification. Each selected training prompt is thus paired with a compact, verified document set $\mathcal{DS}_{\text{cache}}(x)$ indexed by a BM25 retriever. Using a pre-caching strategy, we may not capture all possible information during training, but including relevant documents for each instance ensures a high chance that retrieved evidence will reveal contradictions in incorrect model outputs.

Verification without Claim Decomposition. Instead of extracting and verifying individual claims (as done in VeriScore), we detect contradictions by comparing the entire response with the retrieved documents in a single LM forward pass. This avoids repeated document processing and greatly reduces computation compared to concurrent work using VeriScore as a factuality reward (Chen et al., 2025). Binary RAR achieves a 2×–4× throughput improvement depending on response length, using four replicas of Qwen3-32B as the verifier on a cluster of 8 NVIDIA H100 GPUs.

4 Experimental Setup

4.1 Benchmarking the Hallucination–Utility Trade-off

We curate an evaluation suite that includes four datasets for *hallucination evaluation* and ten datasets for *utility evaluation*, spanning math, code, general chat, and instruction following. Our objective is to *minimize hallucination errors while avoiding performance degradation on utility benchmarks* relative to the original LM.

Hallucination Evaluation We assess hallucination behavior in both long-form generation and short-form question answering using the following datasets: BIOGRAPHY (Min et al., 2023) and WILDHALLUCINATION (Zhao et al., 2024) for long-form generation, and POPQA (Mallen et al., 2023) and GPQA (Rein et al., 2024) for short-form question answering that requires substantial factual knowledge. We report the *hallucination rate* as the primary metric, following the definition used in OpenAI (2025). For long-form generation, the hallucination rate is computed as the proportion of incorrect claims among all extracted atomic claims, which is equivalent to one minus the factual precision in FactScore (Min et al., 2023). We use `gpt-4.1` to extract claims with a customized prompt, retrieve the top 10 document chunks (each 100 words) associated with the prompt entity, and use `gpt-4.1-mini` to verify whether each claim is supported by the retrieved evidence. For short-form QA, we explicitly instruct the model to answer with "I don't know" when uncertain. The hallucination rate is measured as the percentage of incorrect answers. On POPQA, the model produces short answers that are judged by `gpt-4.1` as correct, incorrect, or abstaining. On GPQA, we perform exact matching against the correct multiple-choice option or the "I don't know" string.

Utility Evaluation We evaluate the retention of general utility after continued finetuning. For knowledge retention, we revisit POPQA and GPQA under a no-abstention setup, where the model is prompted to provide an answer (i.e., make its best guess). Accuracy is measured against the ground-truth answers using the same judging method as in the hallucination evaluation. Beyond factual knowledge, we test broader capabilities on eight additional benchmarks: ALPACAEVAL (Dubois et al., 2024), ARENAHARD (Li et al., 2025), and IFEVAL (Zhou et al., 2023) for instruction following; BBH (Suzgun et al., 2023), GSM8K (Cobbe et al., 2021), and MINERVA (Lewkowycz et al., 2022) for reasoning; and HUMANEVAL (Chen et al., 2021) and MBPP (Austin et al., 2021) for code generation. We follow each benchmark's official evaluation protocol. Full details are provided in Appendix B.

4.2 Dataset Curation

Curating high-quality and diverse prompts is essential for effective RL training (Kimi-Team et al., 2025). We aim to reduce hallucination across diverse knowledge domains and instruction types by using natural prompts that reflect realistic user interactions. We build upon WildChat (Zhao et al., 2024), a large collection of natural instruction–response pairs from human interactions with OpenAI models. From this dataset, we automatically identify examples whose responses contain verifiable factual content. We use the OpenAI `gpt-4.1` model with a detailed classification prompt to select suitable examples (see Appendix D).

4.3 Baselines

We compare our method against diverse non-RL and RL baselines with different reward signals. For non-RL methods, we apply supervised fine-tuning (SFT) and direct preference optimization (DPO) to the base reasoning models (Tian et al., 2024; Lin et al., 2024; Chen et al., 2025). For each model, we generate eight responses and evaluate their factuality using the VeriScore pipeline.[1] Specifically, we extract verifiable claims from each response, verify them against pre-cached documents, and compute the percentage of correct claims. For SFT, we fine-tune on the most factual response per instance. For DPO, we construct preference pairs using the two responses with the largest factuality gap and a length difference below 10%, to prevent "length hacking" (Chen et al., 2025).

For RL-based baselines, we consider different reward functions. We first use LM Judge, which rates overall response quality on a 0–10 scale, following common practice (Gunjal et al., 2025). We also test VeriScore (Song et al., 2024) as an RL reward, following concurrent work (Chen et al., 2025). To compute VeriScore, we apply BM25 for retrieval, split documents into 256-token chunks (using the Qwen3 tokenizer), and retrieve the top 4 chunks per claim for verification. Both claim extraction and verification use Qwen3-32B.

4.4 Training Details

We perform continual RL fine-tuning on Qwen3-8B and Qwen3-4B, two reasoning LMs. GRPO serves as the main RL algorithm. We use Qwen3-32B as the verifier to compute binary RAR, prompting it to identify contradictions between model responses and retrieved documents. The learning rate is set to 1×10^{-6}, with KL coefficients of 1×10^{-3} for Qwen3-8B and 3×10^{-3} for Qwen3-4B. To compute binary RAR, we use BM25 retrieval with documents chunked into 512 tokens (using the Qwen3 tokenizer). For each response, we retrieve the top 8 chunks and verify the response with Qwen3-32B. We apply early stopping to prevent overtraining that could degrade utility. Specifically, training is stopped if a checkpoint exhibits more than a 10% drop on any utility benchmark.

5 Main Results

5.1 Results on Hallucination Reduction

Table 1 summarizes hallucination rates across long-form generation and short-form question answering. The base Qwen3-8B model exhibits substantial hallucination, producing 61.9% incorrect claims in long-form generation and 60.6% incorrect answers in short-form QA. Qwen3-4B shows even higher hallucination rates, consistent with prior evidence that smaller models retain less factual knowledge (Mallen et al., 2023). Our proposed approach, RL with Binary RAR, achieves the largest hallucination reduction among all methods, surpassing SFT, DPO, and alternative RL variants.

SFT and DPO Provide Limited Hallucination Reduction. SFT and DPO applied to responses with high VeriScore yield only modest improvements in factuality. On Qwen3-8B, hallucination reduction is small for both long-form (SFT: -1.0; DPO: -8.5) and short-form (SFT: -0.4; DPO: -3.4) settings. These methods rely on an *offline* dataset collected once with the base model. Consequently,

[1] We do not apply SFT or DPO with binary RAR because many prompts yield binary (zero or one) rewards, which makes data generation inefficient.

Models	Long-form (Hallucination Rate ↓)			Short-form (Hallucination Rate ↓)		
	BIOGRAPHY	WILDHALLU	**AVG**	POPQA	GPQA	**AVG**
Qwen3-8B	76.2	47.6	61.9	71.2	50.0	60.6
+ SFT	75.3	46.5	60.9	70.4	50.0	60.2
+ DPO	66.9	39.8	53.4	65.2	49.1	57.2
+ RL (LM Judge)	80.4	50.3	65.4	68.8	48.0	58.4
+ RL (VeriScore)	51.7	29.5	40.6	43.6	41.1	42.3
+ RL (Binary RAR)	45.8	29.2	37.5	26.8	28.3	**27.6**
Qwen3-4B	81.9	50.5	66.2	82.2	55.1	68.7
+ SFT	78.9	48.7	63.8	83.8	54.7	69.2
+ DPO	73.4	43.9	58.7	82.6	54.5	68.5
+ RL (LM Judge)	82.6	53.7	68.1	80.4	54.0	67.2
+ RL (VeriScore)	61.1	32.6	46.9	73.0	51.3	62.2
+ RL (Binary RAR)	46.5	28.9	37.7	46.6	37.3	**41.9**

Table 1: Factuality results comparing different training methods on long-form generation and short-form question answering tasks. We report FactScore precision for long-form generation and hallucination rate for short-form question answering. Binary RAR achieves the best hallucination reduction, showing the highest factual precision and the lowest hallucination rate in short-form question answering. (Revision Note: we replace factual precision with hallucination rate for long-form generation tasks.)

Models	Instruction Following			Knowledge		Reasoning			Coding		
	ALPACA-EVAL	ARENA-HARD	IFEVAL	POPQA	GPQA	BBH	GSM8K	MINERVA	HUMAN-EVAL	MBPP	**AVG**
Qwen3-8B	54.7	18.7	87.2	20.2	48.2	62.4	92.8	80.7	83.5	67.4	61.6
+ SFT	55.7	17.4	86.9	20.4	47.9	59.4	91.6	82.0	83.8	67.0	61.2
+ DPO	53.0	18.3	84.5	18.6	47.5	62.3	90.8	82.1	86.7	67.8	61.2
+ RL (LM Judge)	55.0	18.0	82.2	19.2	52.2	63.1	88.1	77.7	83.8	66.3	60.6
+ RL (VeriScore)	42.2	14.9	88.7	19.6	47.7	61.4	92.2	79.0	83.4	66.9	59.6
+ RL (Binary RAR)	53.9	17.9	85.2	20.6	48.8	66.4	93.4	82.3	86.1	67.6	**62.2**
Qwen3-4B	41.7	12.6	86.1	16.4	44.2	60.9	91.1	82.8	85.5	65.7	58.7
+ SFT	41.2	8.2	82.6	15.2	43.5	59.6	91.4	83.6	83.2	65.6	57.4
+ DPO	39.6	11.0	81.9	15.8	44.0	63.7	90.1	82.7	85.8	66.3	58.1
+ RL (LM Judge)	42.3	11.5	74.3	16.0	43.5	58.1	87.0	82.1	85.9	66.2	56.7
+ RL (VeriScore)	38.4	11.7	86.0	15.4	40.8	59.1	90.8	82.5	84.5	66.2	57.5
+ RL (Binary RAR)	43.0	12.5	84.7	16.4	42.6	58.5	90.7	83.8	84.6	65.0	**58.2**

Table 2: General capability results across ten benchmarks covering instruction following (ALPACAEVAL, ARENAHARD, IFEVAL), knowledge (POPQA, GPQA), reasoning (BBH, GSM8K, MINERVA), and coding (HUMANEVAL, MBPP). We color each cell based on the relative change compared to the base model, where deeper red indicates larger degradation.

factual errors remain in both SFT labels and DPO preferred sequences even after the model evolves, limiting their effectiveness.

Binary RAR Outperforms Other RL Rewards. Among all RL-based approaches, Binary RAR delivers the most consistent and substantial reduction in hallucination. On Qwen3-8B, it lowers long-form hallucination from 61.9 to 37.5 (-24.4) and short-form from 60.6 to 27.6 (-33.0). On Qwen3-4B, hallucination rates drop from 66.2 to 37.7 (long-form) and from 68.7 to 41.9 (short-form), outperforming all baselines. Binary RAR's discrete factual reward penalizes any incorrect content regardless of phrasing or verbosity, preventing reward hacking and maintaining general response quality. By contrast, RL with the continuous VeriScore reward achieves moderate factuality improvement (long-form: -21.3; short-form: -18.3) but remains unstable due to sensitivity to output style and verifier noise. Optimizing for a general LM-judge reward further increases long-form hallucination (65.4), suggesting that optimizing for broad instruction-following or stylistic quality can conflict with factual accuracy.

Models Learn Abstention Behavior. A notable emergent pattern is that RL training encourages models to abstain when uncertain. In short-form question answering, 20%-50% of responses that were previously incorrect are replaced by "I do not know," while correct responses are largely preserved. In long-form generation, models explicitly acknowledge uncertainty about specific entities or facts. We analyze these abstention strategies in detail in §6.2.

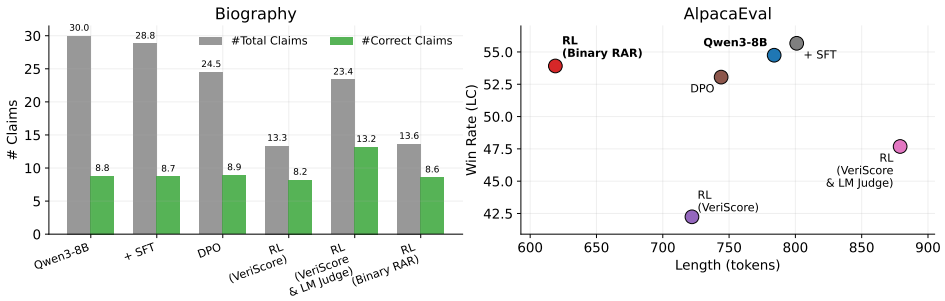

Figure 2: Informativeness in long-form generation. Left: On BIOGRAPHY, Binary RAR cuts the total number of claims but keeps correct claims nearly the same, showing selective filtering of uncertain content. Right: On ALPACAEVAL, Binary RAR gives shorter answers with similar win rates, showing it stays concise without losing quality.

5.2 RESULTS ON GENERAL CAPABILITIES PRESERVATION

Table 2 reports performance across ten benchmarks spanning instruction following, knowledge retention, reasoning, and coding. Binary RAR not only reduces hallucination but also best preserves general capabilities. On Qwen3-8B, RL with Binary RAR achieves an average score of 62.2 , matching the base model's 61.6. In contrast, RL with VeriScore shows clear degradation (59.6).

Open-Ended Chat is Sensitive to Hallucination Reduction. We find that ALPACAEVAL and ARENAHARD are the most sensitive benchmarks to hallucination reduction methods. Both use an LM judge to approximate human preference for long-form outputs, capturing aspects such as relevance, helpfulness, and completeness of the generated responses. When trained with VeriScore-based RL, the model shows substantial performance drops on ALPACAEVAL (54.7→42.2) and ARENAHARD (18.7→14.9). This degradation suggests that continuous rewards such as VeriScore are prone to reward hacking, where the model over-optimizes the proxy signal at the cost of overall response quality. In contrast, RL with Binary RAR preserves scores on these benchmarks, indicating stronger robustness against such overfitting. We analyze this behavior in more detail in § A.1.

Knowledge Retention Despite Abstention. To test whether abstention behavior corresponds to knowledge loss, we evaluate models in a no-abstention setup, where they must always provide an answer. Binary RAR maintains or slightly improves accuracy (POPQA: 20.2→20.6; GPQA: 48.2→48.8), showing that abstention reflects improved uncertainty calibration rather than forgetting factual knowledge.

Reasoning and Coding Remain Intact. Across reasoning and coding benchmarks, all methods show minimal performance change. This stability likely arises because the factuality-oriented training data contains little overlap with these domains, and success in math or code tasks mainly depends on structured reasoning rather than factual recall.

6 ANALYSIS

We next analyze why Binary RAR improves factuality without degrading utility. We examine changes in output informativeness (§6.1), abstention mechanisms (§6.2), and ablation study (§6.3).

6.1 INFORMATIVENESS IN LONG-FORM GENERATION

Although RL with Binary RAR appears to make model outputs less verbose, a closer examination reveals that the informativeness of correct content remains largely unchanged. Figure 2 (left) shows that on the BIOGRAPHY dataset, the total number of claims decreases from 30.0 to 13.6 after Binary RAR training, yet the number of correct claims remains nearly constant (8.8→8.6). This indicates that the model does not simply drop details or shorten text indiscriminately. Instead, it selectively filters out uncertain statements while preserving confident and factually supported information. In other words, the reduction in hallucination arises from improved selectivity rather than content loss.

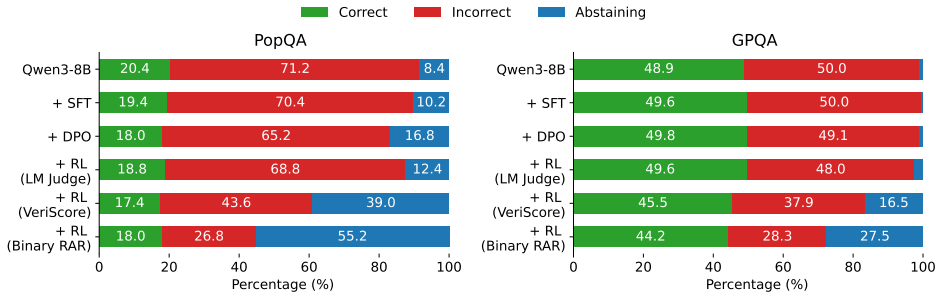

Figure 3: Abstention behavior in short-form question answering. Binary RAR leads the model to abstain on uncertain questions rather than producing incorrect answers, preserving accuracy for attempted ones.

A similar pattern holds when examining the length and win rate on AlpacaEval. As shown in Figure 2 (right), the Binary RAR model generates shorter responses but maintains a comparable win rate. Its length-controlled win rate (54.7→53.9) and vanilla win rate (59.4→59.3) remain mostly unchanged. This suggests that Binary RAR learns to produce more concise yet equally effective outputs and avoids unnecessary verbosity while maintaining the same level of perceived helpfulness and informativeness.

6.2 Abstention Behavior

Recall that we evaluate short-form question answering under two settings: one that allows abstention, used for hallucination evaluation (§5.1), and another that requires forced responses, used for utility evaluation (§5.2). In the hallucination evaluation, we further categorize the answers into three types: correct, incorrect, and abstaining, as shown in Figure 3. The Qwen3-8B model exhibits high error rates and rarely abstains, even on questions it fails to answer correctly. After Binary RAR training, the model's behavior changes substantially: it abstains on 55.2% of PopQA and 27.5% of GPQA questions. Although the overall accuracy slightly decreases (less than a 15% relative reduction), these abstentions are not random. The model primarily abstains on questions it would otherwise answer incorrectly. For questions it attempts to answer, accuracy increases from 22.3% to 40.2% on PopQA and from 49.4% to 60.9% on GPQA. This indicates that the model strategically chooses to abstain when uncertain rather than refusing to answer arbitrarily.

In the standard binary reward design for short-form question answering tasks, a score of one is assigned only when the answer is correct, while zero is given when it is incorrect or expresses uncertainty. In contrast, binary RAR assigns a score of one when the answer is correct *or* when the model explicitly expresses uncertainty, and zero when the answer is incorrect. Since we continue training from a fully post-trained model such as Qwen3, the initial checkpoint already has the capacity to express uncertainty in its output space. Our reward design leverages this ability by encouraging the model to use uncertainty expressions instead of producing incorrect answers. Empirically, with a moderate KL penalty, the model maintains the accuracy of the base model. This outcome arises because the simplest way to maximize reward while minimally altering the model's behavior is to preserve correct answers when confident and express uncertainty when uncertain.

6.3 Ablation Studies

KL Regularization Trade-off. The KL coefficient β controls the balance between reward optimization and staying close to the base model. Figure 4 (left) reveals a critical failure mode at low β values: the model exploits the binary RAR by producing overly short responses. When $\beta = 10^{-3}$, the model maximizes reward by generating brief, uninformative outputs that trivially reduce hallucination rates but degrade the win rate on AlpacaEval. This behavior demonstrates that low KL penalties enable reward hacking. When β is increased to 3×10^{-3}, the stronger constraint to the base model forces the system to maintain informativeness, preventing degenerate solutions and preserving both factuality and general capability.

Early Stopping We study how training duration interacts with the KL coefficient and how these factors influence the balance between factuality and utility. The results in Table 3 show that

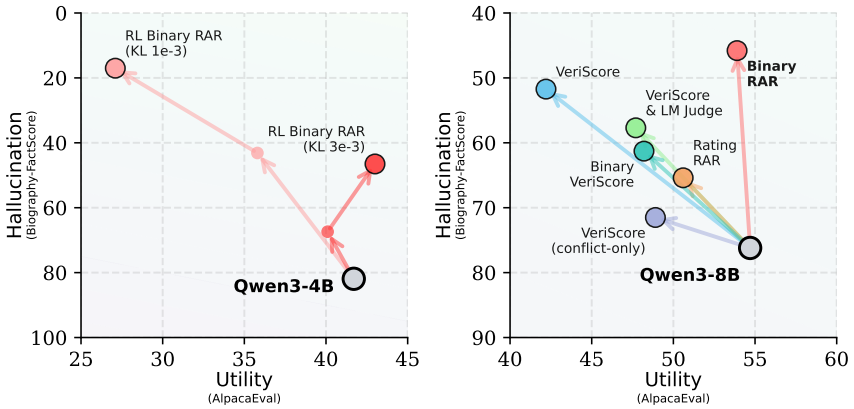

Figure 4: Hallucination–utility tradeoff scatter plot for ablations on different KL coefficients (left) and reward designs (right).

VeriScore improves factual precision rapidly at small KL values, but this comes with a sharp drop in AlpacaEval win rate. At $\beta = 0.001$, extending VeriScore training from 500 to 1000 steps increases Biography precision from 48.3 to 55.6, but win rate falls from 42.2 to 11.6. Raising the KL coefficient to 0.003 improves stability at early steps but weakens hallucination reduction. In contrast, Binary RAR continues to increase factual precision up to 2000 steps while preserving utility. These findings indicate that early stopping partially mitigates the utility collapse for VeriScore, but the trade-off between stability and hallucination reduction remains. Binary RAR does not suffer from this issue, reaching its best performance without harming general capability.

Reward	KL	Steps	Biography (Hallucination Ratio↓)	AlpacaEval (LC Win Rate↑)
VeriScore	0.001	500	51.7	42.2
VeriScore	0.001	1000	44.4	11.6
VeriScore	0.003	500	59.3	45.0
VeriScore	0.003	1000	50.3	33.1
Binary RAR	0.001	1000	58.8	53.7
Binary RAR	0.001	2000	45.8	53.9

Table 3: Ablation on KL coefficient and training steps.

Reward Signal Design. We evaluate three alternative reward schemes to justify the design choices in binary RAR (Figure 4, right). *Binary VeriScore*: Thresholding VeriScore at 0.5 converts the continuous reward into binary form. However, this variant remains sensitive to output style, leading to degraded utility. *Conflict-only VeriScore*: Using the percentage of non-contradictory claims as the reward instead of supported claims. This approach reduces noise from retrieval errors since all responses receive the same reward if all retrieved documents are irrelevant. However, the model exploits this reward by producing less relevant but factually correct statements, lowering AlpacaEval performance. *Rating-based RAR*: Replacing the binary score with a 0–10 factuality rating from the same LM verifier. This design removes dependence on the claim extraction system, but the model exploits the verifier's bias toward certain response styles. Therefore, the effectiveness of binary RAR arises from evaluating the response as a whole and using a binary correctness reward.

7 Conclusion

We present a reinforcement learning fine-tuning approach using a binary retrieval-augmented reward (RAR) to mitigate hallucinations in large language models. By verifying outputs against retrieved evidence and assigning a simple binary score, binary RAR proves more effective than SFT, DPO, or RL with dense rewards such as VeriScore. RL with binary RAR enables models to reduce factual errors in long-form generation, abstain when uncertain in short-form question answering, and at the same time retain knowledge memorization, maintain informativeness, and preserve general capabilities. These results demonstrate that simple binary rewards offer a practical, robust, and scalable path toward safer and more reliable language models.

ETHICS STATEMENT

This research aims to mitigate extrinsic hallucinations in language models, which is crucial for developing safer and more reliable AI systems that users can trust. By improving the factual accuracy of model outputs, this work helps reduce the potential for spreading misinformation. The methods employed use publicly available data and focus on enhancing factual correctness without intentionally introducing new societal biases or risks.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models, 2021. URL https://arxiv.org/abs/2108.07732.

Yejin Bang, Ziwei Ji, Alan Schelten, Anthony Hartshorn, Tara Fowler, Cheng Zhang, Nicola Cancedda, and Pascale Fung. HalluLens: LLM hallucination benchmark. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24128–24156, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1176. URL https://aclanthology.org/2025.acl-long.1176/.

Faeze Brahman, Sachin Kumar, Vidhisha Balachandran, Pradeep Dasigi, Valentina Pyatkin, Abhilasha Ravichander, Sarah Wiegreffe, Nouha Dziri, Khyathi Chandu, Jack Hessel, Yulia Tsvetkov, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. The art of saying no: Contextual noncompliance in language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=f1UL4wNlw6.

Aaron Chatterji, Thomas Cunningham, David J Deming, Zoe Hitzig, Christopher Ong, Carl Yan Shan, and Kevin Wadman. How people use chatgpt. Working Paper 34255, National Bureau of Economic Research, September 2025. URL http://www.nber.org/papers/w34255.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Xilun Chen, Ilia Kulikov, Vincent-Pierre Berges, Barlas Oğuz, Rulin Shao, Gargi Ghosh, Jason Weston, and Wen tau Yih. Learning to reason for factuality, 2025. URL https://arxiv.org/abs/2508.05618.

Qinyuan Cheng, Tianxiang Sun, Xiangyang Liu, Wenwei Zhang, Zhangyue Yin, Shimin Li, Linyang Li, Zhengfu He, Kai Chen, and Xipeng Qiu. Can AI assistants know what they don't know? In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=girxGkdECL.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass. Lookback lens: Detecting and mitigating contextual hallucinations in large language models

using only attention maps. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1419–1436, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.84. URL https://aclanthology.org/2024.emnlp-main.84/.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

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

Yann Dubois, Percy Liang, and Tatsunori Hashimoto. Length-controlled alpacaeval: A simple debiasing of automatic evaluators. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=CybBmzWBX0.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. RARR: Researching and revising what language models say, using language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16477–16508, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.910. URL https://aclanthology.org/2023.acl-long.910/.

Yuzhe Gu, Wenwei Zhang, Chengqi Lyu, Dahua Lin, and Kai Chen. Mask-DPO: Generalizable fine-grained factuality alignment of LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=d2H1oTNITn.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*, 2025.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2), January 2025. ISSN 1046-8188. doi: 10.1145/3703155. URL https://doi.org/10.1145/3703155.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12), March 2023a. ISSN 0360-0300. doi: 10.1145/3571730. URL https://doi.org/10.1145/3571730.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. Towards mitigating LLM hallucination via self reflection. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 1827–1843, Singapore, December 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.123. URL https://aclanthology.org/2023.findings-emnlp.123/.

Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. Why language models hallucinate, 2025. URL https://arxiv.org/abs/2509.04664.

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

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=i1uGbfHHpH.

Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=IFXTZERXdM7.

Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. The dawn after the dark: An empirical study on factuality hallucination in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10879–10899, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.586. URL https://aclanthology.org/2024.acl-long.586/.

Lei Li, Yekun Chai, Shuohuan Wang, Yu Sun, Hao Tian, Ningyu Zhang, and Hua Wu. Tool-augmented reward modeling. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=d94x0gWTUX.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=KfTf9vFvSn.

Yuxin Liang, Zhuoyang Song, Hao Wang, and Jiaxing Zhang. Learning to trust your feelings: Leveraging self-awareness in llms for hallucination mitigation, 2024. URL https://arxiv.org/abs/2401.15449.

Sheng-Chieh Lin, Luyu Gao, Barlas Oguz, Wenhan Xiong, Jimmy Lin, Wen tau Yih, and Xilun Chen. FLAME : Factuality-aware alignment for large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=zWuHSIALBh.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546/.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12076–12100, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.741. URL https://aclanthology.org/2023.emnlp-main.741/.

Benjamin Newman, Abhilasha Ravichander, Jaehun Jung, Rui Xin, Hamish Ivison, Yegor Kuznetsov, Pang Wei Koh, and Yejin Choi. The curious case of factuality finetuning: Models' internal beliefs can improve factuality, 2025. URL https://arxiv.org/abs/2507.08371.

OpenAI. Gpt-5 system card. https://openai.com/index/gpt-5-system-card/, August 2025. Accessed: 2025-10-06.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. LLMs know more than they show: On the intrinsic representation of LLM hallucinations. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=KRnsX5Em3W.

Siya Qi, Lin Gui, Yulan He, and Zheng Yuan. A survey of automatic hallucination evaluation on natural language generation, 2025. URL https://arxiv.org/abs/2404.12041.

Qwen-Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

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

Linxin Song, Taiwei Shi, and Jieyu Zhao. The hallucination tax of reinforcement finetuning, 2025. URL https://arxiv.org/abs/2505.13988.

Yixiao Song, Yekyung Kim, and Mohit Iyyer. VeriScore: Evaluating the factuality of verifiable claims in long-form text generation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 9447–9474, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.552. URL https://aclanthology.org/2024.findings-emnlp.552/.

Zhe Su, Xuhui Zhou, Sanketh Rangreji, Anubha Kabra, Julia Mendelsohn, Faeze Brahman, and Maarten Sap. AI-LieDar : Examine the trade-off between utility and truthfulness in LLM agents. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 11867–11894, Albuquerque,

New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.595. URL https://aclanthology.org/2025.naacl-long.595/.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824/.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. Fine-tuning language models for factuality. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=WPZ2yPag4K.

Bingbing Wen, Jihan Yao, Shangbin Feng, Chenjun Xu, Yulia Tsvetkov, Bill Howe, and Lucy Lu Wang. Know your limits: A survey of abstention in large language models. *Transactions of the Association for Computational Linguistics*, 13:529–556, 2025. doi: 10.1162/tacl_a_00754. URL https://aclanthology.org/2025.tacl-1.26/.

Tianyi Wu, Jingwei Ni, Bryan Hooi, Jiaheng Zhang, Elliott Ash, See-Kiong Ng, Mrinmaya Sachan, and Markus Leippold. Balancing truthfulness and informativeness with uncertainty-aware instruction fine-tuning, 2025. URL https://arxiv.org/abs/2502.11962.

Hongshen Xu, Zichen Zhu, Situo Zhang, Da Ma, Shuai Fan, Lu Chen, and Kai Yu. Rejection improves reliability: Training LLMs to refuse unknown questions using RL from knowledge feedback. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=lJMioZBoR8.

Zijun Yao, Yantao Liu, Yanxu Chen, Jianhui Chen, Junfeng Fang, Lei Hou, Juanzi Li, and Tat-Seng Chua. Are reasoning models more prone to hallucination?, 2025. URL https://arxiv.org/abs/2505.23646.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. R-tuning: Instructing large language models to say 'I don't know'. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7113–7139, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.394. URL https://aclanthology.org/2024.naacl-long.394/.

Wenting Zhao, Tanya Goyal, Yu Ying Chiu, Liwei Jiang, Benjamin Newman, Abhilasha Ravichander, Khyathi Chandu, Ronan Le Bras, Claire Cardie, Yuntian Deng, and Yejin Choi. Wildhallucinations: Evaluating long-form factuality in llms with real-world entity queries, 2024. URL https://arxiv.org/abs/2407.17468.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

A Additional Analysis

A.1 Qualitative Analysis

To better understand the impact of RL training with Binary RAR, VeriScore, and the LM Judge, we present a qualitative analysis of the reward signals and the resulting fine-tuned models.

LM Judge Alone Provides Limited Factuality Assessment. Figure 5 presents two responses to the same instruction along with their evaluations from all three reward models. While the first response contains a factual error and the second is entirely correct, all three rewards appropriately assign lower scores to the erroneous response. However, the LM Judge prioritizes detailed elaboration over factual correctness. When the factual error in the first response is corrected, the Judge only increases its score by 0.1, suggesting that it values comprehensive coverage more than accuracy. This limitation highlights why the LM Judge alone is insufficient for ensuring factuality.

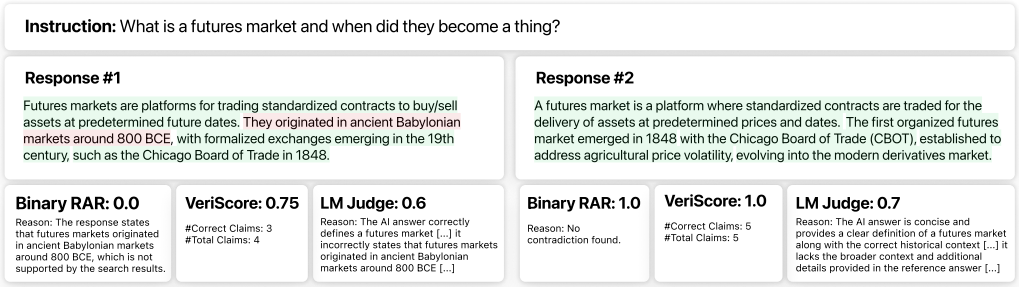

Figure 5: Comparing Binary RAR, VeriScore, and LM Judge on LM responses. Each reward provides a score and reason, with green and red highlights marking correct and incorrect facts; Binary RAR penalizes factual errors most reliably.

VeriScore is Vulnerable to Reward Hacking. As a continuous reward function, VeriScore can incentivize behaviors that conflict with human preferences. Specifically, models can exploit VeriScore in two ways: (1) by generating irrelevant information that is factually correct, and (2) by producing high-level, trivially true statements rather than informative details. Figure 6 illustrates this behavior through examples from models trained with Binary RAR versus VeriScore. The model trained with Binary RAR produces well-structured outputs that contain many details, while the VeriScore-trained model tends to generate more superficial, higher-level descriptions. This demonstrates that continuous reward signals, while well-intentioned, can lead to undesirable failure modes.

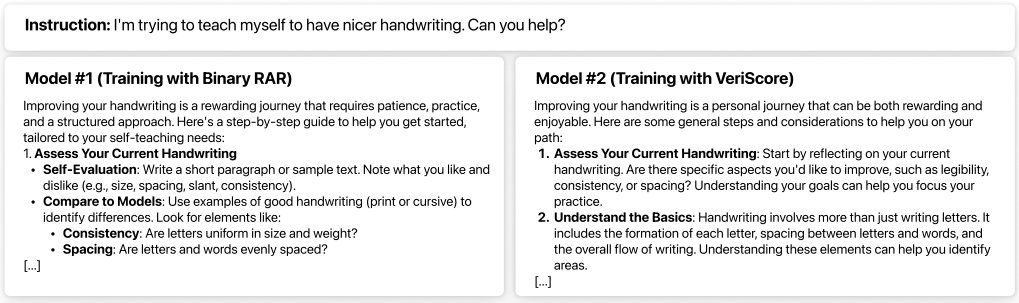

Figure 6: Comparing model outputs trained with Binary RAR and VeriScore. The Binary RAR model gives detailed, structured guidance, while the VeriScore model produces slightly high-level text.

Binary RAR Reduces Hallucination While Preserving Detail. Figure 7 compares outputs from Qwen3-8B before and after RL fine-tuning with Binary RAR. The base model generates incorrect information about Connecticut and Rhode Island, whereas the fine-tuned model avoids these errors

while adding relevant examples of states named after royalty. This demonstrates that RL fine-tuning with Binary RAR effectively reduces factual errors without sacrificing informative content—a crucial advantage over the alternatives explored above.

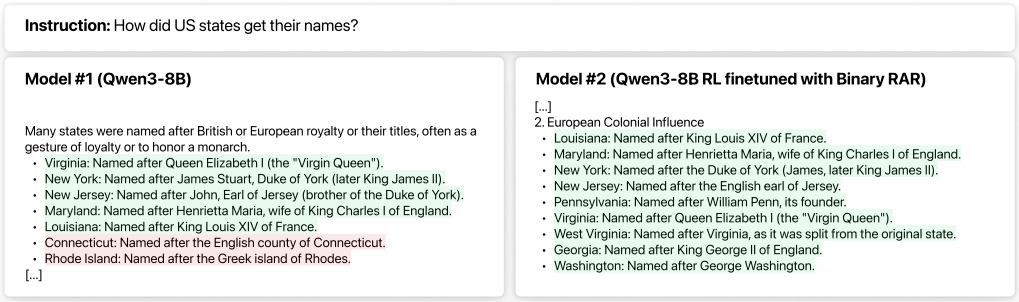

Figure 7: Comparing Qwen3-8B before and after RL fine-tuning with Binary RAR. The fine-tuned model corrects factual errors and keeps relevant details, showing Binary RAR reduces hallucination without losing details.

A.2 Analysis of the Verifier

Since binary rewards depend on detecting contradictions in sampled responses, the verifier's strength directly controls how much useful learning signal the model receives. Table 4 compares Qwen3-8B and Qwen3-32B as verifiers by tracking the percentage of training prompts whose eight sampled responses receive either all-zero rewards, all-one rewards, or a mix. Qwen3-32B identifies contradictions more accurately, producing more mixed-reward cases early in training (27%) compared to Qwen3-8B (19%). Mixed-reward prompts are the only ones that contribute non-zero gradients. As training proceeds, the proportion of all-one cases rises, indicating that the policy produces fewer contradictory outputs. The larger verifier thus supplies more informative gradients and leads to stronger improvements. This pattern does not rule out the usefulness of small verifiers, and prior work has shown that small fine-tuned models can match the accuracy of proprietary systems in fact verification, but in our setting the 32B verifier yields more reliable training signals.

Verifier	all-zero	all-one	mixed
Qwen3-8B	7%→5%	74%→80%	19%→15%
Qwen3-32B	12%→6%	61%→76%	27%→18%

Table 4: Percentage of training prompt categories in the first and last 100 steps.

A.3 Additional Model Families

To assess whether Binary RAR generalizes to other model families, we apply it to Tulu3-8B, which is built on top of the Llama-3.1-8B checkpoint and already trained with a reward model. Table 5 shows that even in this strong starting point, Binary RAR consistently improves factual precision on Biography and WildHallu. Utility metrics remain nearly unchanged, confirming that the method adds factuality without disturbing the model's strengths. This result supports the broader conclusion that Binary RAR is compatible with advanced preference-tuned models and brings reliable gains without inducing undesirable behavior.

Model	Biography (Hallucination Ratio↓)	WildHallu (Hallucination Ratio↓)	AlpacaEval (LC Win Rate↑)	IFEval (Accuracy↑)
Tulu3-8B (base)	68.1	43.4	24.3	81.1
+ Binary RAR	60.1	39.3	23.4	80.6

Table 5: RL with Binary RAR on Tulu3-8B.

B Evaluation Details

B.1 Datasets

We assess hallucination in both long-form generation and short-form question answering using the following benchmarks:

- Biography (Min et al., 2023): A benchmark consisting of prompts that ask models to write biographies of specific individuals.
- WildHallucination (Zhao et al., 2024): A dataset probing factual consistency across diverse real-world entities, including people, geography, and computing, with emphasis on rare entities.
- PopQA (Mallen et al., 2023): A short-form QA dataset covering entities of varying popularity. The correctness is judged automatically by a `gpt-4.1`.
- GPQA (Rein et al., 2024): A multiple-choice QA dataset covering graduate-level biology, chemistry, and physics, where questions and answers are expert-authored.

To measure whether factuality improvements cause regressions in other areas, we evaluate general capabilities using these benchmarks:

- AlpacaEval (Dubois et al., 2024): We use version 2 (v2) and report the length-controlled win rate metric to reduce length bias. The LM judge is `gpt-4.1`.
- ArenaHard (Li et al., 2025): We use version 2.0 and report the style-controlled score. To ensure fair comparison, we add all baselines and our method to the official leaderboard and recompute the regression for style control.
- IFEval (Zhou et al., 2023): A benchmark of 500 prompts covering 25 types of verifiable instructions, designed to test instruction fidelity with objectively checkable outcomes.
- GSM8K (Cobbe et al., 2021): A dataset of grade-school math word problems requiring multi-step reasoning.
- Minerva (Lewkowycz et al., 2022): A collection of 272 graduate-level quantitative reasoning problems in STEM fields such as physics and chemistry, requiring domain-specific expertise.
- HumanEval (Chen et al., 2021): We use HumanEval+, an augmented version of HumanEval that adds additional test cases to improve robustness. Each problem includes multiple functional tests.
- MBPP (Austin et al., 2021): We use BMPP+, an augmented version of MBPP where each instance is equipped with more test cases.

B.2 Abstention Detection

For PopQA, we use an LM judge to classify each output as correct, incorrect, or abstaining. Since PopQA uses short answers, abstentions appear in a very direct form. For GPQA, which is multiple choice, we explicitly prompt the model to output either one option or the exact phrase "I don't know." We manually inspecte 50 outputs from each dataset and find that we were able to correctly identify abstentions in nearly all cases.

C Training Details

RL Fine-tuning. We fine-tune models using reinforcement learning for up to four epochs, with a batch size of 16 unique prompts and 8 rollouts per prompt. Training typically runs for 2,000 steps, except for dense VeriScore rewards, where early stopping at 1,000 steps prevents degradation on utility benchmarks.

SFT and DPO Baselines. For supervised fine-tuning (SFT), one epoch provides the best balance between stability and performance. Direct preference optimization (DPO) is trained for four epochs with factuality-driven preference pairs.

D REWARD IMPLEMENTATION

Data Curation. We curate instruction–response pairs from the WILDCHAT dataset (Zhao et al., 2024) and filter examples with verifiable factual content using `gpt-4.1` with a prompt shown in Figure 12. For each prompt, we pre-cache retrieved documents using the Google Search API, retaining up to 10 relevant web pages. Instances with fewer than three reliable documents are discarded to ensure verification quality. The final curation yields diverse, factual prompts spanning entities, events, and scientific concepts.

Reward Computation. Each instruction–response pair (x, y) is scored by comparing the response against retrieved documents using a verifier LM. The reward is binary:

$$r(x, y) = \begin{cases} 1, & \text{if no contradictions are found between } (x, y) \text{ and retrieved evidence,} \\ 0, & \text{otherwise.} \end{cases}$$

We use Qwen3-32B as the verifier with BM25 retrieval over 512-token chunks (Qwen3 tokenizer). Eight documents are retrieved per instance. This simple binary signal avoids partial credit and reduces noise from verifier bias. For efficiency, each prompt's retrieved set is pre-cached to reduce online retrieval overhead.

Reward Prompts. Figures 8 and 9 show the full prompts used for binary and rating-based retrieval-augmented rewards. These templates define the scoring logic, consistency rules, and JSON output structure for the verifier.

Binary Retrieval-augmented Reward

You are a professional fact-checker tasked with evaluating a language model's response for accuracy. Your job is to systematically compare the response against the provided web search results to identify any factual errors or contradictions. A response deserves a full score if every claim it makes is factually correct, even if it doesn't include every available detail. Omitting information is not a factual error. If a response contains anything that contradicts the world knowledge shown in the search results, it should be considered an error and get a low score.

EVALUATION PROCESS:
1. Read the search results thoroughly to understand the factual baseline
2. Examine each factual claim in the language model's response
3. Cross-reference each claim against the search results
4. Classify each discrepancy according to the guidelines below

CONSIDER CORRECT (No Factual Error):
- Paraphrasing: Same facts expressed in different words
- Reasonable inferences: Logical conclusions drawn from search results
- Partial information: Incomplete but accurate subsets of available information
- Contextual additions: Background information that doesn't contradict search results
- Minor formatting differences: Different ways of presenting same data

CONSIDER INCORRECT (Factual Error):
- Direct contradictions: Response states opposite of what search results show
- Numerical errors: Wrong dates, statistics, quantities, percentages
- Categorical errors: Wrong classifications, locations, names, titles
- Causal errors: Incorrect cause-and-effect relationships
- Timeline errors: Events placed in wrong chronological order
- Attribution errors: Wrong sources, authors, or speakers cited

SCORING RULES:
Your evaluation will result in a binary score: 0 or 1.
- SCORE 1 (No Contradiction): Assign this score if:
1. The response is fully supported by the document.
2. The response contains information that is NOT in the document, but DOES NOT contradict it.
- SCORE 0 (Contradiction): Assign this score ONLY if you find a clear, factual contradiction between the response and the supporting document. A contradiction occurs when the response states the opposite of what the document says (e.g., wrong dates, names, events, or outcomes).

OUTPUT FORMAT:
You must respond ONLY in a valid JSON format with exactly these two fields:
- "REASONING": A brief explanation for your score.
- For SCORE 0, specify the contradiction (e.g., "The response states the event was in 2022, but the document says it was in 2023.").
- For SCORE 1, simply state "No contradiction found."
- "SCORE": An integer, either 0 or 1.

>>> Begin of search results <<<
{passages_text}
<<< End of search results >>>

>>> Begin of the prompt <<<
{prompt_text}
<<< End of the prompt >>>

>>> Begin of the response <<<
{response_text}
<<< End of the response >>>

Respond in JSON format. {{"REASONING": "[...]", "SCORE": "<your-score>"}}

Figure 8: Prompt for binary fact-checking.

Rating Retrieval-augmented Reward

You are a professional fact-checker tasked with evaluating a language model's response for accuracy. Your job is to systematically compare the response against the provided web search results to identify any factual errors or contradictions. A response deserves a full score if every claim it makes is factually correct, even if it doesn't include every available detail. Omitting information is not a factual error. If a response contains anything that contradicts the world knowledge shown in the search results, it should be considered an error and get a low score.

EVALUATION PROCESS:
1. Read the search results thoroughly to understand the factual baseline
2. Examine each factual claim in the language model's response
3. Cross-reference each claim against the search results
4. Classify each discrepancy according to the guidelines below

CONSIDER CORRECT (No Factual Error):
- Paraphrasing: Same facts expressed in different words
- Reasonable inferences: Logical conclusions drawn from search results
- Partial information: Incomplete but accurate subsets of available information
- Contextual additions: Background information that doesn't contradict search results
- Minor formatting differences: Different ways of presenting same data

CONSIDER INCORRECT (Factual Error):
- Direct contradictions: Response states opposite of what search results show
- Numerical errors: Wrong dates, statistics, quantities, percentages
- Categorical errors: Wrong classifications, locations, names, titles
- Causal errors: Incorrect cause-and-effect relationships
- Timeline errors: Events placed in wrong chronological order
- Attribution errors: Wrong sources, authors, or speakers cited

CONFIDENCE SCORING GUIDE:
- 0-2: Very confident there is a factual error (multiple clear contradictions)
- 3-4: Moderately confident there is a factual error (one clear contradiction)
- 5: Uncertain (ambiguous evidence or unclear from search results)
- 6-7: Moderately confident there is no factual error (mostly accurate with minor concerns)
- 8-10: Very confident there is no factual error (all stated facts are accurate, regardless of completeness)

OUTPUT FORMAT REQUIREMENTS:
Respond ONLY in valid JSON format with exactly these two fields:
- "REASONING": A concise explanation of your assessment (1-2 sentences max, e.g., "the response states ... but the search results show ... so there is a factual error" or "no factual error found")
- "SCORE": An integer from 0-10 representing your confidence level

>>> Begin of search results <<<
{passages_text}
<<< End of search results >>>

>>> Begin of the prompt <<<
{prompt_text}
<<< End of the prompt >>>

>>> Begin of the response <<<
{response_text}
<<< End of the response >>>

Respond in JSON format. {{"REASONING": "[...]", "SCORE": "<your-score>"}}

Figure 9: Prompt for rating-based fact-checking.

Claim Extraction for VeriScore Training / FactScore Evaluation

Extract as many fine-grained, atomic, and verifiable factual claims as possible from the response. Each claim should be a single piece of information that could be looked up in a database, official documentation, reputable forum, or reliable source such as Wikipedia or scientific literature.

Guidelines for atomic claims:
- Split a sentence that joins different facts using "and," "or," or by listing into multiple claims.
- If a claim could be split into multiple smaller, independent statements, do so.
- Replace pronouns (e.g., "he", "she", "it", "they") with the full entity name explicitly stated in the response. If the entity name is not explicitly mentioned, leave the pronoun unchanged.
- Extract claims EXACTLY as stated, even if the information appears incorrect or false.

Include as claims:
- Statements about the existence, property, function, or relationship of entities, organizations, concepts, or technologies.
- Claims about names, definitions, features, purposes, or histories.
- Statements about what something does, who runs it, what it is used for, or what it affects.
- For hedged language ("may be," "might be," "could be"), extract the factual association, typical usage, or commonly reported function as long as the claim is traceable to community consensus, documentation, or reputable user reports.
- If a quotation is present, extract it verbatim with the source if given.
- Claims must stand alone, using names or clear descriptions, not pronouns.

Do not include as claims:
- Personal opinions, suggestions, advice, instructions, or experiences.
- Pure speculation or possibilities that are not reported in any documentation or user discussions.
- Claims from code blocks or pure math derivations.

Extract claims only from the response section, not from the prompt or question. If the response does not contain any verifiable factual claims, output an empty list.

Output a JSON list of strings. Each string should be a single atomic factual claim from the response, clearly stated and verifiable.

>>> Begin of prompt <<<
{prompt_text}
<<< End of prompt >>>

>>> Begin of response <<<
{response_text}
<<< End of response >>>

Facts (as a JSON list of strings):

Figure 10: Prompt for atomic claim extraction.

Claim Verification for VeriScore Training / FactScore Evaluation

You need to judge whether a claim is supported or contradicted by Google search results, or whether there is no enough information to make the judgement. When doing the task, take into consideration whether the link of the search result is of a trustworthy source.

Below are the definitions of the three categories:

Supported: A claim is supported by the search results if everything in the claim is supported and nothing is contradicted by the search results. There can be some search results that are not fully related to the claim.
Contradicted: A claim is contradicted by the search results if something in the claim is contradicted by some search results. There should be no search result that supports the same part.
Inconclusive: A claim is inconclusive based on the search results if:
- a part of a claim cannot be verified by the search results,
- a part of a claim is supported and contradicted by different pieces of evidence,
- the entity/person mentioned in the claim has no clear referent (e.g., ”the approach”, ”Emily”, ”a book”).

>>> Begin of search results <<<
{passages_text}
<<< End of search results >>>

Claim: {claim_text}
Task: Given the search results above, is the claim supported, contradicted, or inconclusive? Your answer should be either ”supported”, ”contradicted”, or ”inconclusive” without explanation and comments.

Your decision:

Figure 11: Prompt for claim verification.

Dataset Curation

You need to judge whether a claim is supported or contradicted by Google search results, or whether there is no enough information to make the judgement. When doing the task, take into consideration whether the link of the search result is of a trustworthy source.

Below are the definitions of the three categories:

Supported: A claim is supported by the search results if everything in the claim is supported and nothing is contradicted by the search results. There can be some search results that are not fully related to the claim.
Contradicted: A claim is contradicted by the search results if something in the claim is contradicted by some search results. There should be no search result that supports the same part.
Inconclusive: A claim is inconclusive based on the search results if:
- a part of a claim cannot be verified by the search results,
- a part of a claim is supported and contradicted by different pieces of evidence,
- the entity/person mentioned in the claim has no clear referent (e.g., ”the approach”, ”Emily”, ”a book”).

>>> Begin of search results <<<
{passages_text}
<<< End of search results >>>

Claim: {claim_text}
Task: Given the search results above, is the claim supported, contradicted, or inconclusive? Your answer should be either ”supported”, ”contradicted”, or ”inconclusive” without explanation and comments.

Your decision:

Figure 12: Prompt for dataset curation.