BEYOND MATH AND CODE: LIGHTWEIGHT CORPUS-GROUNDED PROCESS REWARDS FOR FACTUAL QUESTION ANSWERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Process supervision during reinforcement learning (RL) post-training matters in factual question answering (QA) because responses receiving positive response-level rewards can still contain sentence-level factual errors. Unlike math and code, factual QA lacks inexpensive programmatic checks, making reward computation a bottleneck. Existing methods rely on neural verifiers to score individual sentences, requiring extensive model inference as RL repeats these checks across many sampled responses at every update. We therefore introduce **CorVer**[1] (*Corpus Verify*), a lightweight training-time approach to process supervision that derives sentence-level rewards from corpus co-occurrence statistics. A 0.5B extractor identifies subject–object pairs, indexed corpus queries supply their co-occurrence counts, and the resulting sentence rewards are assigned to the corresponding tokens for RL. On Qwen3-4B and Qwen3-8B, CorVer reduces mean complete training time by 5.5–10.4$\times$ relative to the four factuality-RL baselines. Across the four models evaluated against these baselines, CorVer achieves the highest accuracy in 17 of 20 model–benchmark settings. CorVer outperforms the unmodified models in all 30 standard factual-QA settings (six models from three families across five benchmarks) and remains effective on two additional multi-hop QA datasets.

1 INTRODUCTION

Reinforcement learning (RL) post-training of large language models (LLMs) often rewards correct final answers (DeepSeek-AI et al., 2025), even when the reasoning is flawed. Incorporating process supervision into RL provides rewards for intermediate steps (Lightman et al., 2023), but repeated verification can make reward computation a training bottleneck. For example, 100 updates with 3 questions, 16 responses per question, and an assumed five sentences per response entail 24,000 verifier evaluations. Programmatic checks can make verification inexpensive in math and code, but factual question answering (QA) lacks such simple checks for the factual correctness of individual sentences (Figure 2). Existing methods commonly rely on neural verifiers, such as LLM-as-judge models, to compute process rewards (Wu, 2025). Compared with these programmatic checks, LLM-based verification requires substantial inference time for each sentence, making repeated evaluation across sampled responses and RL updates expensive. Moreover, the response generator and the judge are both LLMs, and similar pretraining data may leave them with shared knowledge gaps. The judge may therefore fail to detect factual errors arising from these gaps.

These limitations motivate exploring cheaper supervision signals. Prior work links corpus co-occurrence to factual recall (Kang & Choi, 2023) and uses it for inference-time retrieval (Min et al., 2025). In a manual audit of 700 sentences from QA responses, factual correctness rises monotonically from 24% at zero subject–object co-occurrences to 81% at counts of at least 20 (Figure 1). These counts provide external information for sentence-level rewards without relying on a judge's factual recall.

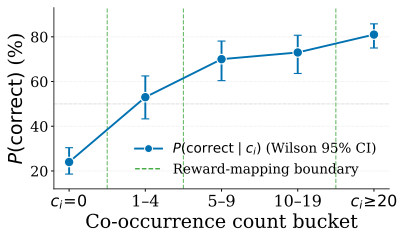
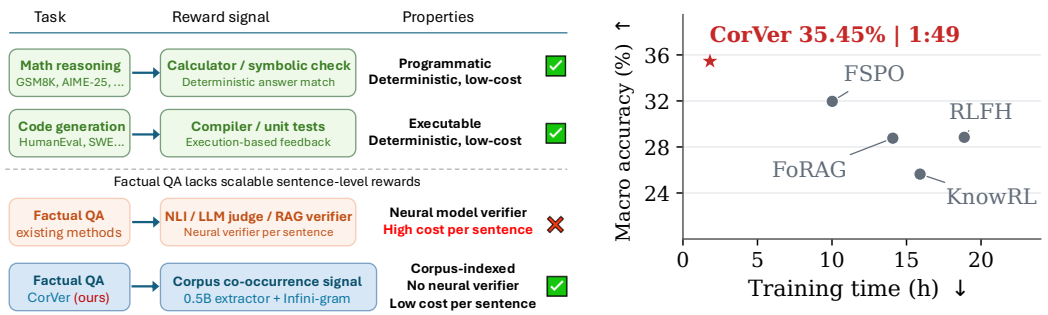


Figure 1: **Counts track factuality.**

[1]Code: `https://anonymous.4open.science/r/CorVer-2EE4`.

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



(a) Reward design

(b) Accuracy and training cost

Figure 2: **Corpus rewards for accurate and efficient factual QA.** (a) Math and code admit inexpensive programmatic checks, whereas existing factual-QA methods rely on neural verification for sentence-level rewards. CorVer instead uses a corpus-indexed co-occurrence signal without a neural verifier. NLI: natural language inference; RAG: retrieval-augmented generation. (b) Mean five-benchmark accuracy and complete training time at step 100 over Qwen3-4B and Qwen3-8B on one H100 80GB. Training configurations and timing details: Appendix D.5.

These informative signals can also be obtained efficiently through indexed corpus queries. For example, Infini-gram reports approximately 20 ms per n-gram count query on the 1.4-trillion-token RedPajama corpus (Liu et al., 2024). This efficiency supports dense sentence-level rewards without repeated neural verification. Because corpus counts are noisy, we combine these rewards with response-level correctness supervision: corpus rewards guide individual sentences, while the correctness reward keeps training directed toward accurate final answers.

We therefore propose **CorVer**, a lightweight approach to process supervision during RL post-training for closed-book factual QA. A 0.5B extractor identifies subject–object pairs, and an Infini-gram corpus index (Liu et al., 2024) supplies their co-occurrence counts. CorVer incorporates these corpus rewards into RL alongside response-level correctness and format rewards. The extractor and corpus index are used only to compute rewards during training. At inference, the model answers using only the question, without corpus access. Figure 2(b) shows that CorVer combines higher accuracy with substantially shorter training time on Qwen3-4B and Qwen3-8B (Yang et al., 2025).

Our main contributions are:

- We introduce a corpus-based process reward that maps subject–object co-occurrence counts to sentence-level rewards for factual-QA post-training, without requiring a neural verifier.
- We demonstrate consistent accuracy gains across six models and five factual-QA benchmarks, with further gains after training on two multi-hop QA datasets. FActScore evaluation also shows more factually supported reasoning on Llama-3.2-3B-Instruct.
- We show that CorVer reduces complete training time by 5.5–10.4× relative to FSPO (Li & Ng, 2025), RLFH (Wen et al., 2025), FoRAG (Cai et al., 2024), and KnowRL (Ren et al., 2025) on Qwen3-4B and Qwen3-8B while achieving higher mean accuracy.

2 Method

CorVer trains a factual-QA policy by combining sentence-level corpus rewards with response-level correctness and format rewards. Figure 3 illustrates reward construction and policy optimization.

2.1 Response Representation

Let x denote a factual question and π_θ the policy. The policy receives a fixed instruction and the question. Reference answers and aliases enter only the reward computation. We sample G completions per question, with $G = 16$ in our experiments. A completion $y = (y_1, \ldots, y_T)$ is prompted to enclose its reasoning in `<think>...</think>` and its final answer in `<answer>...</answer>` tags. After stripping tags, we split the reasoning and final answer separately into sentences $s_1, \ldots, s_m$. The

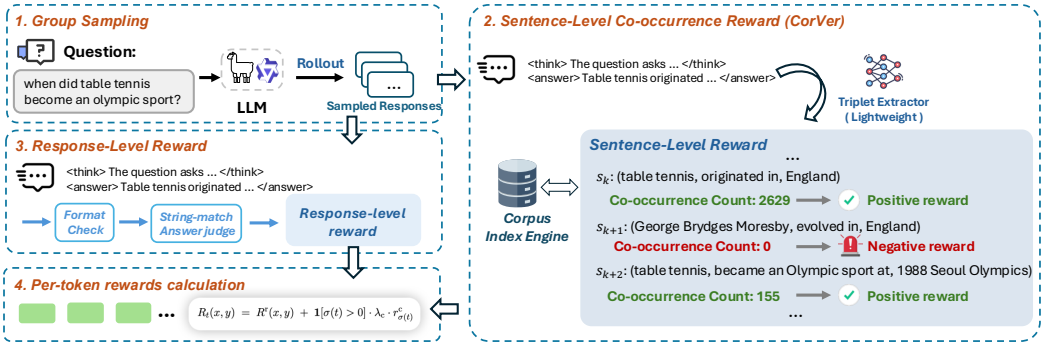


Figure 3: **CorVer pipeline.** Each sentence is scored for corpus co-occurrence via an Infini-gram index. The per-sentence reward is mapped to token-level returns through σ and combined with response-level signals in a policy-gradient update using group relative policy optimization (GRPO).

alignment $\sigma : \{1, \ldots, T\} \to \{0, 1, \ldots, m\}$ maps each token to its sentence, with $\sigma(t) = 0$ for tags and inter-sentence whitespace. This alignment assigns sentence-level rewards to the corresponding tokens, while response-level rewards apply throughout the completion. Training details appear in §3.

2.2 Sentence-Level Co-occurrence Reward

CorVer scores each sentence based on how often its subject and object co-occur in a reference corpus. We first extract $(\text{head}, \text{relation}, \text{tail})$ triplets using a frozen 0.5B extractor. We retain the first valid triplet and use its head and tail as the entity pair, discarding the relation. A valid pair has nonempty, non-pronominal entities whose content words form a query with at least two distinct terms. Appendix C.2 describes this FIRST rule, its failure handling, and comparisons with MIN and RELCHECK. For sentence s_i, let e_i^{s} and e_i^{o} denote the retained head and tail strings. We reduce each entity to its content words to accommodate surface-form variation across the corpus. The union of these words forms a word-level AND query to the co-occurrence index, giving the count

$$c_i \;=\; \mathrm{count}_{\mathcal{W}}(w_1 \wedge \cdots \wedge w_k)\,, \tag{1}$$

where $\mathcal{W}$ is the reference corpus (a fixed Wikipedia snapshot in our experiments) and $w_1, \ldots, w_k$ are the distinct content words derived from $(e_i^{\mathrm{s}}, e_i^{\mathrm{o}})$. The Infini-gram index (Liu et al., 2024) counts corpus positions where the query terms co-occur within a fixed token-distance bound. Appendix A.4 provides the query and extractor settings. We map this count to a sentence-level reward:

$$r_i^{\mathrm{c}} \;=\; \begin{cases} \alpha_0, & c_i = 0, \\ \alpha_1, & 0 < c_i < \tau_1, \\ \alpha_2, & \tau_1 \le c_i < \tau_2, \\ \alpha_3, & c_i \ge \tau_2, \\ 0, & c_i = \bot, \end{cases} \tag{2}$$

where $\alpha_0 < \alpha_1 \le \alpha_2 < \alpha_3$ are bounded reward levels and $\tau_1 < \tau_2$ are integer thresholds. We set $c_i = \bot$ when no valid entity pair or usable count is available, distinguishing missing evidence from a successful query with zero matches. The observation in Figure 1 motivates using counts as a weak factuality signal (Appendix B). The bounded mapping supplies sentence rewards alongside correctness supervision. Section 3 gives reward values and thresholds.

2.3 Response-Level Rewards

CorVer combines the sentence-level signal with two response-level rewards. The *correctness reward* $R^{\mathrm{j}}(x, y)$ evaluates only the final answer outside the reasoning block through deterministic string and alias matching. Incorrect, refused, and empty answers receive the same penalty. The *format reward* $R^{\mathrm{f}}(y)$ checks for a nonempty, closed final-answer block and includes a linear penalty for excess completion length. Reward values and format rules appear in Appendix A.1.3.

2.4 CORPUS REWARDS AND TOKEN ADVANTAGES

For completion $y^{(g)}$, let $r^{\mathrm{c}}_{g,i}$ denote the reward of sentence i. The alignment σ_g assigns this reward to the corresponding tokens. With $r^{\mathrm{c}}_{g,0} = 0$, the token return is

$$R_{g,t} = \underbrace{\lambda_{\mathrm{j}} R^{\mathrm{j}}(x, y^{(g)}) + \lambda_{\mathrm{f}} R^{\mathrm{f}}(y^{(g)})}_{\text{response-level supervision}} + \underbrace{\lambda_{\mathrm{c}} \mathbf{1}[\sigma_g(t) > 0]\, r^{\mathrm{c}}_{g,\sigma_g(t)}}_{\text{sentence-level corpus supervision}}. \tag{3}$$

The weights $\lambda_{\mathrm{j}}, \lambda_{\mathrm{f}}, \lambda_{\mathrm{c}}$ control the three reward channels. The first two terms form the response-level return R^{r} shown in Figure 3. Tokens in either the reasoning or answer block receive the reward of their sentence, while tags and inter-sentence whitespace receive only response-level supervision.

We normalize token returns using group statistics computed from each completion's mean return:

$$\bar{R}_g = \frac{1}{T_g} \sum_{t=1}^{T_g} R_{g,t}, \qquad A_{g,t} = \frac{R_{g,t} - \mu_x}{s_x + \varepsilon}, \tag{4}$$

where T_g counts valid completion tokens, excluding padding, and μ_x and s_x are the mean and standard deviation of $\{\bar{R}_g\}_{g=1}^{G}$ for question x, with $\varepsilon = 10^{-4}$. The numerator retains sentence-level reward differences, so sentences within a completion can receive different advantages. Setting $\lambda_{\mathrm{c}} = 0$ yields a shared response-level advantage within each completion. We use these advantages in clipped GRPO with Kullback–Leibler (KL) regularization against the frozen initial base model (Shao et al., 2024). Low-rank adaptation (LoRA) (Hu et al., 2021) updates average over valid tokens within each completion and then across the group (Appendix C.3).

Efficient implementation. Sentences from multiple completions are processed together, repeated queries reuse cached counts, and the resulting rewards are assigned back to each completion's tokens. Extraction runs in a persistent worker with CUDA Graph execution, and token-reward tensors are constructed on the central processing unit (CPU) and transferred to the GPU in batches. Appendix A.4 lists the corpus query parameters and sampling settings for the extractor.

3 EXPERIMENTAL SETUP

Data and model. We use TriviaQA (Joshi et al., 2017), NQ-Open (Kwiatkowski et al., 2019), PopQA (Mallen et al., 2023), SimpleQA (Wei et al., 2024), and TruthfulQA (Lin et al., 2022). The main comparison uses Llama-3.2-3B-Instruct, Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Qwen3-4B, and Qwen3-8B (Yang et al., 2025), with additional tests on OLMo-2-13B-Instruct (Team OLMo et al., 2025) and Qwen3-14B. We also use HotpotQA (Yang et al., 2018) and 2WikiMultiHopQA (Ho et al., 2020) for multi-hop QA. We evaluate Llama-3.2-3B-Instruct reasoning on 1,000 HotpotQA validation questions.

Baseline and training. As discussed above, existing methods typically compute sentence-level factuality rewards through NLI verification or LLM-as-judge. FSPO (Li & Ng, 2025), our closest comparator, combines NLI-based sentence rewards with answer correctness. Related methods target complementary goals: RLFH (Wen et al., 2025) balances truthfulness and informativeness, FoRAG (Cai et al., 2024) targets factual retrieval-augmented long-form QA, and KnowRL (Ren et al., 2025) encourages knowledge-boundary awareness and abstention. Their original goals differ from closed-book answer accuracy, so we evaluate their transfer to this setting. Raw is the unmodified model. Training uses single-GPU LoRA and one NQ-Open/WebQuestions (Berant et al., 2013) pool shared across methods and all six base models. CorVer uses GRPO. In Eq. (2), we set $(\alpha_0, \alpha_1, \alpha_2, \alpha_3) = (-0.2, -0.1, 0, +0.1)$ and $(\tau_1, \tau_2) = (5, 20)$. Multi-hop training uses each target dataset separately. We test author-reported hyperparameters (FSPO, RLFH), rollout allocation (FoRAG), and training duration (KnowRL), retaining the stronger complete baseline row by macro accuracy. Appendices A.1, A.3, and D.3 provide training and configuration details.

Evaluation. We report closed-book answer accuracy and its five-benchmark macro average. For multi-hop QA, exact match (EM) requires a normalized reference match, while token-level F1, the

Table 1: Factual QA accuracy (%) on five benchmarks. For FSPO, RLFH, FoRAG, and KnowRL, we retain the complete row with higher macro accuracy where alternative settings were tested. CorVer uses its final step-100 checkpoint. Bold/underline mark the best/second-best scores within each model. Differences are relative to the best competitor (red: higher; green: lower).

Model	Method	TriviaQA	NQ-Open	PopQA	SimpleQA	TruthfulQA	Macro
Llama-3.2-3B-Instruct	Raw	57.84	36.65	18.54	2.24	36.35	30.32
	FoRAG	60.79	40.28	21.17	2.24	42.35	33.37
	RLFH	59.12	36.90	20.75	2.61	37.21	31.32
	FSPO	59.45	40.25	22.11	**4.30**	41.13	33.45
	KnowRL	50.23	29.70	15.17	1.53	27.42	24.81
	CorVer (ours)	**66.19** (+5.40)	**45.48** (+5.20)	**26.26** (+4.15)	3.91 (-0.39)	**46.76** (+4.41)	**37.72** (+4.27)
Llama-3.1-8B-Instruct	Raw	74.67	44.93	31.97	5.69	46.27	40.70
	FoRAG	66.79	33.43	25.73	5.55	36.23	33.55
	RLFH	72.49	41.22	29.92	5.36	38.92	37.58
	FSPO	79.81	51.02	**36.77**	**6.89**	47.61	44.42
	KnowRL	71.14	41.52	29.19	2.43	36.60	36.18
	CorVer (ours)	**81.50** (+1.69)	**53.07** (+2.05)	36.69 (-0.08)	6.70 (-0.19)	**49.45** (+1.84)	**45.48** (+1.06)
Qwen3-4B	Raw	52.32	24.82	17.73	2.96	45.29	28.62
	FoRAG	51.52	24.71	16.93	2.10	39.53	26.96
	RLFH	50.68	24.60	15.99	1.60	36.60	25.89
	FSPO	53.55	26.79	18.14	3.37	44.80	29.33
	KnowRL	44.09	20.69	11.35	0.46	27.42	20.80
	CorVer (ours)	**57.57** (+4.02)	**31.86** (+5.07)	**22.16** (+4.02)	**4.23** (+0.86)	**50.31** (+5.02)	**33.22** (+3.89)
Qwen3-8B	Raw	63.75	30.94	21.45	3.01	42.72	32.37
	FoRAG	60.86	28.42	19.81	2.73	41.00	30.57
	RLFH	62.77	30.39	20.52	2.91	42.35	31.79
	FSPO	64.77	32.24	22.77	3.91	49.33	34.60
	KnowRL	62.19	30.78	18.20	2.45	38.80	30.48
	CorVer (ours)	**67.39** (+2.62)	**36.87** (+4.63)	**27.57** (+4.80)	**4.92** (+1.01)	**51.65** (+2.32)	**37.68** (+3.08)

harmonic mean of token precision and recall, credits partial matches. FActScore-NPM (Min et al., 2023) uses a nonparametric masked language model (NPM) to evaluate reasoning through micro atomic precision, supported facts per response, coverage, and facts per response. Training cost is complete run time on one H100 80GB. Protocols appear in Appendices A.2, A.3, and D.5.

4 Experiments

4.1 Main Results

CorVer leads in 17 of 20 model–benchmark settings and exceeds FSPO, the strongest baseline by macro accuracy, on all four models (Table 1). Both Qwen3 models outperform FSPO on every benchmark, whereas the Llama models show a less uniform advantage. Llama-3.1-8B has the smallest macro margin over FSPO at 1.06 percentage points (pp). These results suggest limited transfer from the related baselines' original objectives to closed-book accuracy. KnowRL illustrates the mismatch: longer Qwen3-4B training increases refusals, which receive no accuracy credit, without improving accuracy (Appendix D.3).

Qualitative analysis of Llama-3.1-8B finds newly correct answers alongside regressions, sometimes involving familiar background knowledge that diverts responses from the requested fact. These losses partly offset the corrections, whereas newly correct answers substantially outnumber regressions in other models. We conjecture that this model's existing knowledge and reasoning behavior are less well matched to corpus rewards, contributing to its smaller advantage over FSPO (§4.2).

4.2 Gains across Model Families and Scales

Figure 4 covers six models from three families, with positive gains in all 30 model–benchmark settings. Mean gains are largest on TruthfulQA (+9.59 pp) and NQ-Open (+9.44 pp), followed by PopQA (+8.06 pp) and TriviaQA (+7.81 pp), and smallest on SimpleQA (+2.29 pp). The task-level pattern also differs between models of similar size: at 8B, Llama gains more on NQ-Open, whereas Qwen3 gains more on TruthfulQA. This reversal suggests that the benefit depends on the combination of model and task, which the macro average alone does not reveal.

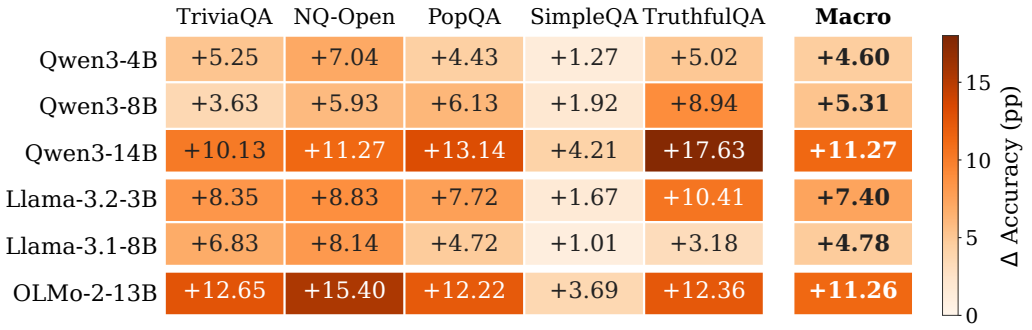


Figure 4: **CorVer accuracy gains over Raw (pp).** Rows group models by family. Macro is the mean gain across the five benchmarks.

The largest gains occur on OLMo-2-13B and Qwen3-14B: both exceed 11 pp, substantially more than any 3B–8B model. Within Qwen3, gains rise from $+4.60$ to $+5.31$ and $+11.27$ pp as scale increases. This pattern is consistent with our hypothesis that a stronger pretrained foundation makes imperfect corpus rewards more useful (§5.3). We conjecture that richer knowledge helps a model select and connect facts it already knows, whereas smaller models may be more susceptible to misleading rewards. The Llama trend differs ($+7.40$ pp at 3B versus $+4.78$ pp at 8B), suggesting that this benefit depends on existing capabilities as well as scale. Appendix D.1 reports the full accuracies.

4.3 Accuracy and Training Cost

Table 2: Accuracy and complete training cost at step 100, averaged over Qwen3-4B and Qwen3-8B. Time ratios are relative to CorVer.

Method	Macro accuracy (%) ↑	Time (h:mm) ↓	Time ratio ↓
CorVer	**35.45**	**1:49**	**1.00**×
FSPO	31.97	10:01	5.52×
RLFH	28.84	18:53	10.41×
FoRAG	28.76	14:05	7.76×
KnowRL	25.64	15:55	8.77×

We report complete training wall-clock time, excluding job queueing and evaluation. This includes application programming interface (API) communication and response waiting, since policy updates require the remote judges' rewards. CorVer gains 3.49 pp over FSPO while taking less than one fifth as long (Table 2). CorVer is also faster than FSPO and FoRAG, whose rewards are computed locally without remote API calls (Appendix D.5).

4.4 Factuality of the Generated Reasoning

Higher answer accuracy does not establish that the reasoning itself is more factual. We therefore compare Raw and CorVer using FActScore-NPM, which assesses support for atomic facts: individual, checkable claims extracted from the reasoning. Atomic precision is the fraction of extracted facts supported by the evaluator's evidence, pooled across responses. Supported facts per response measures the amount of supported content, while coverage is the fraction of responses with at least one scorable fact. Facts per response counts all extracted facts, including unsupported ones.

Table 3: FActScore-NPM for Llama-3.2-3B-Instruct. Precision is micro-averaged.

Metric	Raw	CorVer
Atomic precision (%) ↑	83.73	**84.98**
Supported facts / response ↑	1.292	**1.443**
Coverage (%) ↑	61.40	**67.90**
Facts / response	1.543	1.698

CorVer produces more facts per response while increasing atomic precision (Table 3), indicating that the additional content is accompanied by better factual support. Higher coverage further shows that a larger share of responses contains scorable facts.

4.5 Learning on Multi-Hop QA

Multi-hop QA tests whether sentence-level corpus rewards can improve answers that require combining multiple facts. We compare task-specific CorVer and FSPO training against Raw on HotpotQA and 2WikiMultiHopQA.

Table 4: Closed-book multi-hop QA after dataset-specific training, evaluated by answer EM and F1 (%). Bold marks the best score within each model.

		HotpotQA		2WikiMultiHopQA	
Model	Method	EM ↑	F1 ↑	EM ↑	F1 ↑
Llama-3.2-3B-Instruct	Raw	7.99	14.28	3.03	7.28
	FSPO	14.80	22.00	18.69	22.20
	CorVer	**16.42**	**23.06**	**19.61**	**23.47**
Qwen3-4B	Raw	10.93	18.16	12.53	20.76
	FSPO	17.19	**24.78**	16.86	19.88
	CorVer	**17.85**	24.71	**25.25**	**29.27**
Llama-3.1-8B-Instruct	Raw	14.44	23.69	8.87	17.71
	FSPO	18.87	27.48	24.33	30.01
	CorVer	**22.77**	**30.56**	**24.82**	**30.13**
Qwen3-8B	Raw	19.01	26.51	21.62	26.19
	FSPO	21.11	29.24	19.48	22.72
	CorVer	**21.42**	**29.91**	**27.50**	**32.01**

CorVer achieves higher EM than FSPO in every model–dataset comparison (Table 4), indicating that it produces the complete reference answer more often. Its advantage is especially pronounced on 2WikiMultiHopQA, where both Qwen models gain over 8 pp in EM over FSPO. The F1 results broadly agree: CorVer leads except on Qwen3-4B HotpotQA, where the methods differ by only 0.07 pp. Together, these results show that sentence-level co-occurrence rewards can support training for questions requiring multiple facts, even though the reward does not explicitly verify the links between reasoning steps. Appendix A.3 reports additional Raw–CorVer comparisons on the larger models.

5 Analysis and Discussion

5.1 Contributions of the Reward Components

The ablations separate the reward source from its assignment. We hold the base model fixed at Llama-3.2-3B-Instruct and remove one component at a time. A1 removes corpus rewards, while A2–A4 retain the zero-count penalty of -0.2. Across all five benchmarks, the full configuration outperforms each variant and Raw (Table 5).

Table 5: Ablation accuracy (%) on Llama-3.2-3B-Instruct. Each variant removes the named component. Bold marks the best score on each benchmark.

Variant	TriviaQA	NQ-Open	PopQA	SimpleQA	TruthfulQA
Raw	57.84	36.65	18.54	2.24	36.35
CorVer (full)	**66.19**	**45.48**	**26.26**	**3.91**	**46.76**
A1: − corpus reward (vanilla GRPO; correctness + format rewards)	59.61	39.81	20.10	2.73	40.51
A2: − correctness reward (per-token corpus reward + format only)	59.02	40.06	21.11	2.52	39.78
A3: − per-token alignment (sentence-mean corpus reward)	63.53	43.88	24.67	3.56	41.74
A4: − self-filter (full 9,680-question pool)	63.52	42.96	23.09	2.64	43.33

Corpus rewards and the correctness reward contribute differently across benchmarks. Removing the corpus reward (A1) lowers accuracy on all five, including TriviaQA (66.19 to 59.61) and NQ-Open (45.48 to 39.81). Removing the correctness reward (A2) produces the largest drop on TriviaQA (7.17 pp), SimpleQA (1.39 pp), and TruthfulQA (6.98 pp), whereas removing the corpus reward costs more on NQ-Open (5.67 pp) and PopQA (6.16 pp). Reward assignment also matters. A3 retains the same corpus signal but averages it into a single response score. Accuracy falls on every benchmark, most on TruthfulQA (41.74 versus 46.76), suggesting that the model benefits from rewards tied to individual sentences. Finally, using the unfiltered question pool (A4) lowers accuracy throughout, with the largest drops on TruthfulQA (3.43 pp) and PopQA (3.17 pp).

5.2 Gains across Entity Popularity

We examine how CorVer's PopQA gains vary with subject-entity popularity (Mallen et al., 2023). Figure 5 covers six models in three scale groups (3–4B, 8B, and 13–14B), retaining each model's results within the group. All models use shared popularity quartiles over the full 14,267 questions; Appendix D.2 reports each model's full-set gain and the gains within all four quartiles.

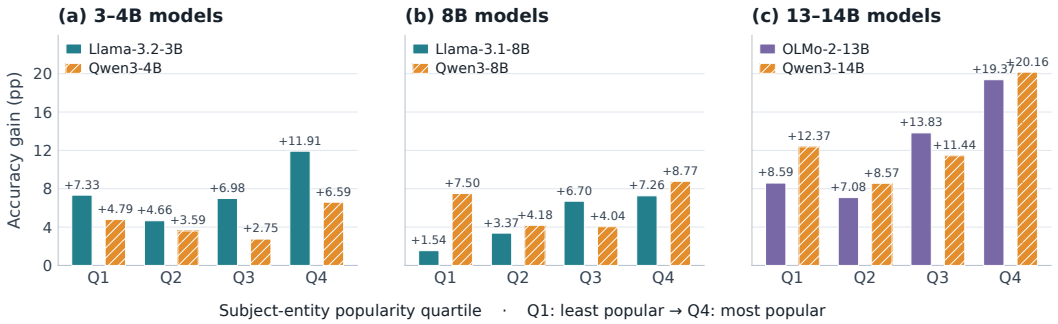


Figure 5: PopQA accuracy gains over Raw (pp) by entity popularity. All models use the same quartiles, from the least popular entities (Q1) to the most popular (Q4).

CorVer achieves its largest gains in Q4 for every model. The equally weighted six-model means are $+12.34$ pp for Q4, $+7.02$ and $+7.62$ pp for Q1 and Q3, and $+5.24$ pp for Q2.

These averages show that CorVer's gains are strongest for popular entities but also substantial for the least popular entities, rather than increasing monotonically with popularity. The ordering varies across models: Qwen3-4B and Qwen3-8B gain less in Q3 than in Q2. We conjecture that low-popularity entities benefit from penalties on unsupported associations, while high-popularity entities benefit from reinforcement of correct associations. At intermediate popularity, co-occurrence rewards may distinguish correct from incorrect associations less clearly, potentially limiting the mean gain in Q2.

5.3 Why Corpus Rewards Help in Post-Training

Corpus counts can penalize rare facts or reward false claims involving familiar entities. Their usefulness in CorVer rests on three considerations. First, our 700-sentence audit is stratified by co-occurrence count (Appendix B). Within these sampled buckets, 81% of sentences with $c_i \geq 20$ are correct, while zero-count sentences are 76% incorrect and 24% correct. These bucket-level proportions support positive and negative rewards as noisy signals, with factual precision increasing monotonically across all five buckets (Figure 1).

Second, response-level correctness rewards complement corpus rewards when frequency and correctness disagree. Correct answers receive a correctness reward even when rare associations incur corpus penalties, while incorrect answers receive a correctness penalty despite frequent entity pairs. Removing either reward source reduces accuracy across all five benchmarks (§5.1), supporting their complementary roles.

Third, we hypothesize that LLM-as-judge supervision may involve a form of circular reasoning. Prior work links factual recall to pretraining exposure and documents difficulties with rarely co-occurring entities (Kandpal et al., 2023; Kang & Choi, 2023). If the policy and judge learn from similar corpora, limited exposure to the same facts may impair both generation and verification. When the judge relies on its parametric knowledge, the potential circularity lies in using one model's uncertain knowledge to validate another's, without independent factual evidence. Under these conditions, LLM-as-judge rewards may not be more reliable than noisy corpus rewards. CorVer queries corpus statistics directly, without requiring a verifier to recall the relevant facts. Although these counts also miscredit some sentences, they provide an external signal to refine a policy with existing factual and reasoning capabilities. KL regularization discourages large departures from the initial policy, which may help preserve useful behavior under noisy rewards. We conjecture that this combination helps explain the observed training gains.

6 Related Work

Process supervision and corpus evidence. RL post-training commonly uses response-level objectives (Ouyang et al., 2022; Shao et al., 2024; DeepSeek-AI et al., 2025), while process supervision evaluates intermediate reasoning (Lightman et al., 2023; Nie et al., 2026). Fine-Grained RLHF (Wu et al., 2023) learns segment-level rewards for reinforcement learning from human feedback (RLHF). Retrieved evidence supports generation in RAG (Lewis et al., 2020) and factuality evaluation in FActScore (Min et al., 2023), while Self-RAG (Asai et al., 2024) combines retrieval with self-critique. Factual recall correlates with co-occurrence and pretraining exposure (Kang & Choi, 2023; Kandpal et al., 2023), and QuCo (Min et al., 2025) uses co-occurrence to guide retrieval. CorVer instead uses Infini-gram counts (Liu et al., 2024) as inexpensive process rewards.

Factuality-aware post-training. FSPO (Li & Ng, 2025) combines sentence-level entailment checks with answer rewards, making it our closest baseline. RLFH (Wen et al., 2025) balances truthfulness and informativeness: its policy judges atomic facts against evidence, with information rewards discouraging uninformative refusals or overly brief answers. FoRAG (Cai et al., 2024) targets factual, coherent long-form answers grounded in retrieved evidence through outline-based generation and fine-grained reward modeling. KnowRL (Ren et al., 2025) aims to keep reasoning within the model's knowledge boundaries, rewarding refusal when uncertain alongside correctness and factuality. Other work studies factuality-oriented online RL (Chen et al., 2025), direct preference optimization (Tian et al., 2024), factuality-aware supervised and preference training (Lin et al., 2024), and refusal-aware instruction tuning (Zhang et al., 2024). CorVer focuses on efficient sentence-level rewards for closed-book QA, combining corpus counts with answer correctness.

7 Limitations and Conclusion

Limitations. Corpus co-occurrence is an imperfect proxy for factuality: frequent entity pairs can occur in false claims, and rare but correct facts can receive penalties. Its usefulness depends on corpus coverage and extraction quality, while systematic mismatches between frequency and correctness may limit further gains. Our evaluation is strictly closed-book: policies receive only the question at inference. CorVer's training-time corpus queries return counts for rewards, without supplying retrieved text to the policy. We do not compare against RAG systems that condition answer generation on retrieved passages.

Conclusion. CorVer turns inexpensive corpus co-occurrence statistics into sentence-level process rewards for factual-QA post-training. Across six models, it improves accuracy in all 30 model–benchmark settings, with additional gains in task-specific multi-hop QA and atomic factual precision. On Qwen3-4B and Qwen3-8B, it achieves higher macro accuracy than all four RL baselines while reducing complete training time by 5.5–$10.4\times$. Ablations support both combining corpus and correctness rewards and preserving rewards at the sentence level. The results establish corpus statistics as a practical source of process rewards for factual QA, reducing the cost of sentence-level supervision while improving both final answers and the factual content of reasoning. The extractor and index are needed only during training, so these gains require no additional verification components at inference.

AI USE STATEMENT

LLM-based tools were used for code and implementation assistance and for manuscript writing and refinement, including editing and LaTeX formatting. The authors reviewed the AI-assisted code and manuscript and take responsibility for the final research design, implementation, analyses, reported results, and paper content.

ETHICS STATEMENT

The manual factuality audit was conducted by the authors; no external annotators, crowdworkers, or personal data were involved.

CorVer aims to improve factuality, but generated answers can still be incorrect. Corpus co-occurrence may reflect biases in source coverage and underrepresent rare entities and facts. Models trained with these rewards should therefore not be treated as authoritative fact-checkers, and their outputs require independent verification in high-stakes applications.

REPRODUCIBILITY STATEMENT

The anonymous repository provides the CorVer training code, configurations for all six base models, the selected training questions, and links to publicly available baseline resources: `https://anonymous.4open.science/r/CorVer-2EE4`. Appendix A details the training, data processing, and evaluation protocols.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi 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://proceedings.iclr.cc/paper_files/paper/2024/hash/25f7be9694d7b32d5cc670927b8091e1-Abstract-Conference.html.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1533–1544, 2013. URL https://aclanthology.org/D13-1160/.

Tianchi Cai, Zhiwen Tan, Xierui Song, Tao Sun, Jiyan Jiang, Yunqi Xu, Yinger Zhang, and Jinjie Gu. FoRAG: Factuality-optimized retrieval augmented generation for web-enhanced long-form question answering. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 199–210, 2024. URL https://doi.org/10.1145/3637528.3672065.

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

DeepSeek-AI et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning, January 2025. URL https://arxiv.org/abs/2501.12948.

Aaron Grattafiori et al. The Llama 3 Herd of Models, November 2024. URL https://arxiv.org/abs/2407.21783.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625. International Committee on Computational Linguistics, 2020. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models, October 2021. URL https://arxiv.org/abs/2106.09685.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, 2017. URL https://aclanthology.org/P17-1147/.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *International conference on machine learning*, pp. 15696–15707. PMLR, 2023. URL https://proceedings.mlr.press/v202/kandpal23a.html.

Cheongwoong Kang and Jaesik Choi. Impact of co-occurrence on factual knowledge of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 7721–7735, 2023. URL https://aclanthology.org/2023.findings-emnlp.518/.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural Questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. URL https://aclanthology.org/Q19-1026/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Junyi Li and Hwee Tou Ng. Reasoning models hallucinate more: Factuality-aware reinforcement learning for large reasoning models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 151064–151085. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/ddd50f29fa472095515fa0df31749e6c-Paper-Conference.pdf.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step, May 2023. URL https://arxiv.org/abs/2305.20050.

Sheng-Chieh Lin, Luyu Gao, Barlas Oğuz, Wenhan Xiong, Jimmy Lin, Wen-tau Yih, and Xilun Chen. FLAME: Factuality-aware alignment for large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/d16152d53088ad779ffa634e7bf66166-Abstract-Conference.html.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 3214–3252, 2022. URL https://aclanthology.org/2022.acl-long.229/.

Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens. *arXiv preprint arXiv:2401.17377*, 2024. URL https://arxiv.org/abs/2401.17377.

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 *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 9802–9822, 2023. URL https://aclanthology.org/2023.acl-long.546/.

Dehai Min, Kailin Zhang, Tongtong Wu, and Lu Cheng. QuCo-RAG: Quantifying uncertainty from the pre-training corpus for dynamic retrieval-augmented generation. *arXiv preprint arXiv:2512.19134*, 2025. URL https://arxiv.org/abs/2512.19134.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12076–12100, 2023. URL https://aclanthology.org/2023.emnlp-main.741/.

Shuo Nie, Hexuan Deng, Chao Wang, Ruiyu Fang, Xuebo Liu, Shuangyong Song, Yu Li, Min Zhang, and Xuelong Li. Stop rewarding hallucinated steps: Faithfulness-aware step-level reinforcement learning for small reasoning models. *arXiv preprint arXiv:2602.05897*, 2026. URL https://arxiv.org/abs/2602.05897.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, March 2022. URL https://arxiv.org/abs/2203.02155.

Baochang Ren, Shuofei Qiao, Da Zheng, Huajun Chen, and Ningyu Zhang. KnowRL: Exploring knowledgeable reinforcement learning for factuality. *arXiv preprint arXiv:2506.19807*, 2025. URL https://arxiv.org/abs/2506.19807v3.

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, April 2024. URL https://arxiv.org/abs/2402.03300.

Team OLMo et al. 2 OLMo 2 Furious, October 2025. URL https://arxiv.org/abs/2501.00656.

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://proceedings.iclr.cc/paper_files/paper/2024/hash/c361ae924c23cafca6033610d25dbc65-Abstract-Conference.html.

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

Xueru Wen, Jie Lou, Xinyu Lu, Yuqiu Ji, Xinyan Guan, Yaojie Lu, Hongyu Lin, Ben He, Xianpei Han, Debing Zhang, and Le Sun. On-policy self-alignment with fine-grained knowledge feedback for hallucination mitigation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5215–5231, 2025. URL https://aclanthology.org/2025.findings-acl.271/.

Xiaobao Wu. A comprehensive survey on learning from rewards for large language models: Reward models and learning strategies. *EMNLP (Findings)*, pp. 17847–17875, 2025.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for language model training. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/b8c90b65739ae8417e61eadb521f63d5-Abstract.html.

An Yang et al. Qwen3 technical report, May 2025. URL https://arxiv.org/abs/2505.09388.

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

Processing, pp. 2369–2380. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

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 *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. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.394. URL https://aclanthology.org/2024.naacl-long.394/.

A Training Setup and Recipe

A.1 Comparison Protocol and Training Settings

All methods are evaluated in closed-book QA: the policy receives only the question at inference. Evidence retrieval and reward models are used during training where required by each method.

A.1.1 Methods and Their Implementations

CorVer. CorVer combines corpus-based sentence rewards with correctness and format rewards. A frozen 0.5B extractor maps sentences to entity pairs, and an Infini-gram index supplies their co-occurrence counts. The resulting rewards are assigned to sentence tokens for GRPO training. The extractor runs on the training GPU, while corpus lookup runs on the CPU.

FSPO. FSPO (Li & Ng, 2025)[2] combines answer correctness with factuality rewards on reasoning steps. A resident Hughes Hallucination Evaluation Model (HHEM-2.1) NLI verifier checks sentences against evidence, and its factuality rewards enter token-level advantages for GRPO training.

RLFH. RLFH (Wen et al., 2025)[3] uses truthfulness and informativeness rewards for proximal policy optimization (PPO) training. Our frozen judge is `Qwen/Qwen3.8-Flash-Next-FP8` (180B total parameters, 6B active), accessed through a managed OpenAI-compatible endpoint. Its prompt and decoding configuration follow the original RLFH setup.

FoRAG. FoRAG (Cai et al., 2024) decomposes generated content into subclaims and verifies them against evidence to compute fine-grained factuality rewards. We implement this reward procedure for closed-book policy training with PPO. Both FoRAG and RLFH maintain a critic and reuse each sampled batch for four PPO epochs.

KnowRL. KnowRL (Ren et al., 2025)[4] combines answer correctness with atomic-fact support to encourage awareness of knowledge boundaries. Our GRPO implementation uses GPT-4o-mini, DeBERTa-v3 NLI, and a Wikipedia knowledge base for reward computation. We retain its reward for appropriate abstention (Appendix D.3.4).

[2] https://github.com/nusnlp/FSPO
[3] https://github.com/AlignRM/RLFH
[4] https://github.com/zjunlp/KnowRL

A.1.2 SHARED EXPERIMENTAL CONDITIONS

The main five-benchmark comparison holds the following conditions fixed across methods for each base model:

- **Training questions.** All methods and all six base models use the same selected pool of 6,621 questions from NQ-Open and WebQuestions, prepared once. Appendix A.2 gives the selection procedure.
- **Adaptation and hardware.** Training uses single-GPU LoRA with rank 128, scaling 256, and seed 42, with one seed per configuration. Baseline adapters use dropout 0 and no bias. Each timed run uses one H100 80GB.
- **Generation limit.** Training rollouts and main-table evaluation responses are limited to 1,000 generated tokens.
- **Evaluation.** Methods share the same question-only system prompt, benchmark questions, answer extraction, scoring rules, and decoding settings for a given base model. Empty answers and refusals receive zero accuracy credit. The prompt, model-specific decoding settings, and scoring rules are given in Appendix A.2.

Table 6: Shared rollout allocations and additional baseline configurations, grouped by model. Within each main model, the 100-step rows with $C = 16$ share the same generation budget. FoRAG (adjusted) and KnowRL (extended) report additional configurations. FSPO and RLFH use their main-table settings. Q is questions per step and C is completions per question. Steps are optimizer updates for GRPO and outer batches for PPO. Responses count newly generated completions ($Q \times C \times$ steps).

Base model	Method / configuration	Q	C	Steps	Responses
Llama-3.2-3B	CorVer	3	16	100	4,800
	FSPO	3	16	100	4,800
	RLFH	3	16	100	4,800
	FoRAG (initial)	3	16	100	4,800
	FoRAG (adjusted)	4	1	100	400
	KnowRL	3	16	100	4,800
	KnowRL (extended)	3	16	200	9,600
Qwen3-4B	CorVer	4	16	100	6,400
	FSPO	4	16	100	6,400
	RLFH	4	16	100	6,400
	FoRAG (initial)	4	16	100	6,400
	FoRAG (adjusted)	4	1	100	400
	KnowRL	4	16	100	6,400
	KnowRL (extended)	4	16	200	12,800
Llama-3.1-8B	CorVer	3	16	100	4,800
	FSPO	3	16	100	4,800
	RLFH	3	16	100	4,800
	FoRAG (initial)	3	16	100	4,800
	FoRAG (adjusted)	4	1	100	400
	KnowRL	3	16	100	4,800
	KnowRL (extended)	3	16	200	9,600
Qwen3-8B	CorVer	3	16	100	4,800
	FSPO	3	16	100	4,800
	RLFH	3	16	100	4,800
	FoRAG (initial)	3	16	100	4,800
	FoRAG (adjusted)	4	1	100	400
	KnowRL	3	16	100	4,800
	KnowRL (extended)	3	16	200	9,600
OLMo-2-13B	CorVer	3	16	100	4,800
Qwen3-14B	CorVer	3	16	100	4,800

A.1.3 METHOD-SPECIFIC SETTINGS AND SELECTION

Table 7: Algorithm, learning rate, KL coefficient, and reward components. CorVer uses KL 0.01 for Qwen3-4B, 0.003 for Llama-3.1-8B, and 0.001 otherwise. RLFH uses KL 0.01 on Llama-3.2-3B and Qwen3-4B, and 0.05 on the two 8B models. Other baseline learning-rate (LR) and KL values apply across the four main models. Configuration comparisons and checkpoint selection are described in Appendix D.3.

Method	Algo.	G	Steps / ckpt.	LR / β_{KL}	Reward-side components
CorVer	GRPO	16	100 / step-100	1×10^{-5} / 0.001	0.5B extractor + Infini-gram lookup
FSPO	GRPO	16	100 / step-100	2×10^{-6} / 0.04	HHEM-2.1 NLI, threshold 0.5
RLFH	PPO	—	100 / step-100	5×10^{-7} / 0.01 or 0.05	Qwen3.8-Flash-Next-FP8 judge
FoRAG	PPO	—	100 / step-100	5×10^{-7} / 0.02	subclaim decomposition + evidence-field verification (Cai et al., 2024)
KnowRL	GRPO	16	100 / step-100	1×10^{-5} / 0.001	GPT-4o-mini judge + DeBERTa-v3 NLI (thr. 0.3) + Wikipedia knowledge base

Learning rates and KL coefficients are tuned empirically using the original papers and available official implementations as references. Tables 6 and 7 give the generation budgets and selected settings. Where alternatives are evaluated, we retain the complete baseline row with higher five-benchmark macro accuracy, with all scores from one checkpoint. PPO epochs reuse responses and do not multiply the generation budget. Complete-run costs are reported in Appendix D.5.

Selected baseline configurations. FSPO uses the adapted configuration at rollout temperature 0.7. RLFH uses author-key settings at 1.0 for Llama-3.2-3B and Qwen3-4B, and adapted settings at 0.7 for the two 8B models. FoRAG uses adjusted rollout allocations for Llama-3.2-3B and Qwen3-8B and initial allocations for the other two models, all at temperature 1.0. KnowRL uses step-100 checkpoints for accuracy and cost comparisons. Appendix D.3 reports the configuration and duration comparisons supporting these choices.

CorVer training and rewards. CorVer trains a fresh LoRA adapter from the official base model without a supervised fine-tuning (SFT) cold start, using rollout temperature 1.0 and the settings in Tables 6 and 7. For Eq. (2), $(\alpha_0, \alpha_1, \alpha_2, \alpha_3) = (-0.2, -0.1, 0.0, +0.1)$ and $(\tau_1, \tau_2) = (5, 20)$ (Appendix B). Correctness gives $+2$ for a matched final answer and -1 for an incorrect, refused, or empty answer. We set $\lambda_{\mathrm{f}} = \lambda_{\mathrm{c}} = 1$ and $\lambda_{\mathrm{j}} = 1$, except $\lambda_{\mathrm{j}} = 2$ for Llama-3.1-8B. Under token averaging, the mean corpus reward lies in $[-0.2, 0.1]$, below the correctness reward swing of $3\lambda_{\mathrm{j}}$. Alternative aggregation rules appear in Appendix C.2.

The format reward checks the final-answer block outside the reasoning block and penalizes excess completion length:

$$R^{\mathrm{f}}(y) = \begin{cases} +1, & \text{the final-answer block is nonempty and closed,} \\ -1, & \text{otherwise} \end{cases} - \eta \frac{\max\{0, L(y) - L_0\}}{L_0}, \quad (5)$$

where $L(y)$ is the completion length in tokens, $\eta = 0.5$, and $L_0 = 512$. Answer correctness is scored separately, so a well-formed incorrect answer can receive the positive format component.

A.2 DATASETS AND EVALUATION PROTOCOL

Evaluation datasets. Each CorVer run evaluates the final checkpoint on five closed-book QA datasets in series: TriviaQA validation (17,944 questions), NQ-Open validation (3,610), PopQA test (14,267), SimpleQA (4,326), and TruthfulQA (817). Each TriviaQA record carries a question, a primary answer, and an alias list.

Training data and selection. Training questions come from NQ-Open and WebQuestions. Preparation removes duplicate questions within the training pool and applies Wikipedia entity grounding. The 6,621-question pool is selected once using Llama-3.1-8B and shared across all methods and all

six base models, without separate selection for each model or method. The budgets in Table 6 count sampled instances, which need not be distinct or cover the pool.

The pool combines learning-zone questions, which received both correct and incorrect sampled answers during selection, with already-mastered anchor questions. Ablation A4 uses the unfiltered candidate pool to examine this selection (§5.1).

Training and evaluation grading. For CorVer's five-benchmark experiments, the training correctness reward uses normalized answer and alias matching on the final answer outside the reasoning block. At evaluation, TriviaQA, NQ-Open, PopQA, and SimpleQA use bidirectional substring matching after normalization: a response matches if its final answer contains a reference answer or alias, or is contained in one. TruthfulQA instead computes ROUGE-1 F1 between the final answer and each reference, taking separate maxima over the correct and incorrect reference sets. An answer is correct only when the maximum for correct references is strictly greater than the maximum for incorrect references. Ties, including two zero maxima, score 0. We report the accuracy of this binary decision, not mean ROUGE-1 F1. Appendix D.4 reports supplementary checks using one-way matching and semantic judgments.

On all five benchmarks, empty answers and explicit refusals score 0, and the denominator retains every test question, including duplicates inherent in the source data. A response counts as an explicit refusal only when its normalized answer span exactly matches a refusal phrase (*I don't know*, *I do not know*, *I have no comment*). Empty answers are distinguished from explicit refusals.

Reasoning-trace factuality evaluation. We compare Llama-3.2-3B-Instruct Raw with CorVer (-0.2, FIRST, final step-100) on the same $1{,}000$ HotpotQA distractor-validation questions, sampled in advance with seed 42. Both receive only the same system prompt and question. Generation uses the Brain Floating Point 16-bit (BF16) format, greedy decoding, a $1{,}000$-token output limit, and a $2{,}048$-token context limit. We score the `<think>` content preceding the first `<answer>`; if think tags are absent, we use the text before the answer block. Final answers are excluded.

We use FActScore-NPM (Min et al., 2023) with independent Wikipedia retrieval and a QA-oriented article-retrieval adaptation. A Qwen3 extractor produces verbatim atomic claims at temperature 0 with thinking disabled; exact duplicates are removed. Article retrieval uses the fixed `enwiki-20230401.db` Wikipedia snapshot. SQLite Full-Text Search 5 (FTS5) retrieves the top article using an OR query over up to 12 deduplicated non-stopwords from the claim, with Best Matching 25 (BM25) title/body weights $5/1$. Official BM25 passage retrieval selects the top three passages within that article. The official `facebook/npm-single` model scores support in 32-bit floating-point (FP32) precision; a fact is supported only if its NPM probability is strictly greater than 0.3. The evaluator uses no HotpotQA context, supporting facts, or gold answers, and applies no γ length penalty. Its retrieval and support scoring are separate from the training co-occurrence lookup.

Factuality metrics. For each model, let $N = 1{,}000$ be the number of responses, F the total atomic facts, S the supported facts, and A the responses with at least one scorable fact. We report micro atomic precision $100S/F$, supported facts per response S/N, coverage $100A/N$, and facts per response F/N. Empty reasoning and fact-free responses remain in N. Coverage measures scorable-response prevalence, and facts per response is a quantity diagnostic. Raw has $(F, S, A) = (1{,}543, 1{,}292, 614)$ and CorVer has $(1{,}698, 1{,}443, 679)$.

Prompting and generation length. All main-table methods use the following system prompt, which requests brief reasoning, a direct answer, the specified abstention format, and no repetition. The user message contains only the original question, without retrieved material or examples. The shared generation limit is $1{,}000$ tokens.

```
Answer using this exact format:
<think>brief reasoning</think>
<answer>direct answer</answer>

If you genuinely do not know, reply exactly: <think>unknown</think><
answer>I don't know</answer>
Do not loop or repeat the same point or phrase.
```

Decoding settings. All methods share evaluation decoding within each base model. Llama-3.2-3B-Instruct and Qwen3-8B use greedy decoding with temperature 0 and repetition penalty 1.0. Llama-3.1-8B-Instruct uses the same settings with repetition penalty 1.02. Its Raw and both FSPO configurations use one beginning-of-sequence (BOS) token per input. Qwen3-4B follows the model card's recommended thinking-mode sampling: temperature 0.6, top-p 0.95, top-k 20, presence penalty 1.5, and repetition penalty 1.0.[5] In the scaling experiments, Qwen3-14B and OLMo-2-13B-Instruct use greedy decoding with temperature 0 and repetition penalty 1.0 for Raw and CorVer. Other baselines are not evaluated on these two models. The shared $1{,}000$-token output limit applies throughout.

A.3 Multi-Hop Dataset-Specific Training and Evaluation

Training data and initialization. For the CorVer and FSPO runs in Tables 4 and 8, each model–dataset run initializes a fresh LoRA adapter from the official base model. All six models use the same target-task pools: $4{,}064$ HotpotQA and $1{,}208$ 2WikiMultiHopQA questions. The pools were selected using a response-level-reward model and are fixed across all six models. We check their official training-split provenance, remove duplicate training questions, and exclude question-identifier or normalized-question overlaps with either official development set. Reference answers enter only the training reward; reference passages, titles, supporting facts, and evidence are not supplied to the policy.

CorVer training configuration. CorVer retains the Llama-3.2-3B co-occurrence, format, and length rewards (Appendix A.1.3). Correctness uses official normalized exact match ($+2$ if correct, -1 otherwise), including official aliases and demonyms for 2WikiMultiHopQA. Runs use LoRA rank/scaling $128/256$, dropout 0, learning rate 10^{-5}, KL coefficient 0.001, AdamW8bit, BF16, 16 generations per question, and seed 42. The additional 4B–14B runs use H100 GPUs. Evaluation uses the final checkpoint after 100 optimizer steps.

FSPO comparison. FSPO is trained separately on both target datasets for the four 3B–8B models, yielding eight runs with the same selected pools as CorVer. All use LoRA rank/scaling $128/256$ and 3 questions with 16 responses each per step, including Qwen3-4B, totaling $4{,}800$ responses over 100 steps. Final step-100 checkpoints are evaluated on each full development set. These four models form the method comparison, while the larger models provide supplementary Raw–CorVer evaluations.

Evaluation protocol. We evaluate the full official development sets: $7{,}405$ HotpotQA and $12{,}576$ 2WikiMultiHopQA questions. Within each model, Raw, CorVer, and FSPO (where evaluated) share the closed-book system prompt and question, with fixed prompt date 07 Sep 2026. Decoding and the $1{,}000$-token output limit follow Appendix A.2. Llama-3.2-3B uses context length $2{,}048$. Official scoring scripts compute answer EM and token-overlap F1, with refused and empty answers scoring zero and all development questions retained in the denominator. Tables 4 and 8 report answer scores, while reasoning FActScore is evaluated separately in §4.4.

Additional results on larger models. Table 8 additionally reports Raw and CorVer results on OLMo-2-13B-Instruct and Qwen3-14B after the same dataset-specific training.

A.4 Implementation Details

Fixed index and extractor. Following Min et al. (2025), we implement the word-level AND query as a conjunctive normal form (CNF) count over corpus token positions using Infini-gram (Liu et al., 2024). The fixed Wikipedia index uses `max_diff_tokens=1000` and `max_clause_freq=500000`. The distance limit is measured in tokens of the fixed Llama-2 index tokenizer; it is an engine distance bound, not a symmetric $\pm 1{,}000$-token window or a document count. The training co-occurrence index is separate from the knowledge base used for NPM evaluation.

We use QuCo-extractor-0.5B (Min et al., 2025), a Qwen2.5-0.5B-Instruct model fine-tuned for triplet extraction, and keep it frozen during policy training. The extractor worker uses BF16, temperature

[5] `https://huggingface.co/Qwen/Qwen3-4B#best-practices`

Table 8: Additional closed-book multi-hop results on larger models (%). CorVer uses the final step-100 checkpoint trained separately for each dataset. Answer EM and F1 cover the full official development sets; CorVer uses a zero-count penalty of -0.2. Bold marks the higher value within each model pair.

		HotpotQA		2WikiMultiHopQA	
Model	Method	EM ↑	F1 ↑	EM ↑	F1 ↑
OLMo-2-13B-Instruct	Raw	20.23	27.73	21.04	24.86
	CorVer	**25.73**	**35.28**	**26.49**	**32.17**
Qwen3-14B	Raw	16.91	22.49	11.73	14.34
	CorVer	**23.07**	**31.50**	**26.18**	**31.26**

0.7, top-p 0.8, top-k 20, and repetition penalty 1.1, with at most 256 new tokens and a total context limit of 4,096.

Efficient reward computation. The policy uses vLLM for rollout generation with eager execution. We batch sentences across sampled completions and extract triplets with a persistent, frozen QuCo-extractor-0.5B worker using CUDA Graph execution on the same GPU. Extracted triplets are mapped back to their original sentences before corpus queries and reward assignment.

A local corpus index and a query-count cache reduce repeated lookups. Successful counts, including zero, are cached, while failed queries are not. We construct token rewards on the CPU using the sentence–token alignment and transfer them to the GPU in batches, excluding padding from the objective. Each sentence reward applies only to its corresponding tokens. Ablation A3 instead broadcasts an averaged sentence reward (§5.1).

At inference, only the trained policy is used, with no extractor or corpus lookup.

B Motivating Audit of Corpus Co-occurrence

We manually examined 700 sentences sampled by co-occurrence bucket from Llama-3.1-8B-Instruct responses to questions drawn from publicly available QA sources (Figure 1). Five of the authors independently labeled the 400 sentences in the zero-count and highest-count buckets after checking claims with web search. Four of the authors labeled the remaining 300 sentences, with external sources available for verification. Final labels were determined by majority vote, with 2–2 ties resolved through discussion. Labels assess sentence factuality independently of final-answer correctness.

C Co-occurrence Reward Design and Analysis

Appendices C.1 and C.2 evaluate reward sensitivity and triplet aggregation on TriviaQA using models trained on the NQ-Open and WebQuestions pool (Appendix A.2). The main reward settings, including the zero-count penalty -0.2, the FIRST rule, and the count thresholds, were selected independently of these evaluations. TriviaQA questions and responses are used exclusively for evaluation and analysis.

C.1 Zero-Count Penalty Sensitivity

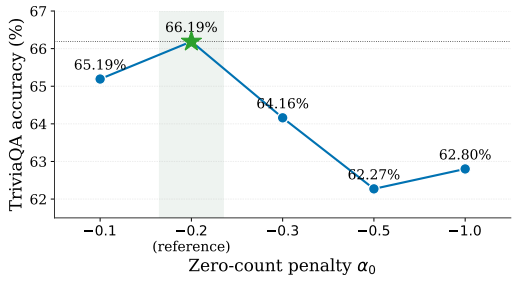


Figure 6: TriviaQA accuracy under five zero-count penalties. The star marks the -0.2 setting, and the dotted line shows its reference accuracy.

We vary only the zero-count penalty α_0 over $\{-0.1, -0.2, -0.3, -0.5, -1.0\}$ using Llama-3.2-3B-Instruct, the FIRST scoring rule, seed 42, and final step-100 checkpoints. Evaluation covers all 17,944 TriviaQA validation questions.

The -0.2 setting reaches 66.19%, the highest accuracy among the five settings. Relative to this reference, -0.1, -0.3, -0.5, and -1.0 reduce accuracy by 1.00, 2.03, 3.92, and 3.39 pp, respectively. Differences are computed from correct-answer counts.

C.2 Triplet Aggregation and Relation Checks

This comparison uses Llama-3.2-3B-Instruct, seed 42, final step-100 checkpoints, and zero-count penalty -0.2. Full uses the FIRST rule and matches the main experiment.

Valid triplets and queries. A valid triplet is a three-element list whose subject and object are nonempty strings outside the implementation's pronoun list and can form an AND query with at least two distinct content terms.

FIRST and MIN. FIRST queries the first valid triplet's subject and object. An unsuccessful query gives reward 0, without trying a later triplet. MIN queries all valid pairs and uses the smallest successful count. Failed queries are omitted, while a successful zero count remains eligible for the minimum. If no usable count is available, the reward is 0.

RELCHECK. RELCHECK starts from FIRST. A subject–object count $c_i \geq 20$ triggers a subject–relation–object query. A successful zero count lowers the reward from $+0.1$ to -0.05. A positive count, a failed query, or an unusable relation query leaves the original reward unchanged. The same content-word extraction is applied to all three fields, and the merged query must contain at least three distinct terms. This check adds relation words to the corpus query rather than applying an entailment model.

Query construction. Content-word extraction first selects capitalized non-stopwords longer than one character, then falls back to non-stopwords longer than two characters, and finally to the original stripped string. Query terms retain their original order and are deduplicated case-insensitively. The shared extractor and these query rules are fixed across the three variants.

Comparison results. On the full TriviaQA validation set, Full (FIRST) achieves 66.19% accuracy, compared with 61.25% for MIN and 61.34% for RELCHECK, corresponding to drops of 4.94 and 4.85 pp. In this comparison, checking more triplets or adding relation terms does not improve answer accuracy. For MIN, sparse coverage of secondary claims may introduce extra penalties.

Table 9: Triplet-aggregation comparison on Llama-3.2-3B-Instruct (TriviaQA validation, $N = 17{,}944$). All runs use seed 42, final step-100 checkpoints, and zero-count penalty -0.2. Differences are computed from correct-answer counts.

Configuration	Accuracy (%)	Δ vs. Full (pp)
Full (FIRST)	**66.19**	0.00
MIN	61.25	−4.94
RELCHECK	61.34	−4.85

C.3 CorVer Algorithm

Algorithm 1 summarizes how CorVer constructs token rewards and advantages. Reference answers are used only for reward computation. The group statistics are computed from the mean token return of each completion, with padding excluded throughout.

Algorithm 1 CorVer: Corpus Rewards with Token Alignment

Require: Policy π_θ, training data $\mathcal{D}$, group size G

for each training iteration **do**
  Sample question x and reference answers from $\mathcal{D}$
  Sample $\{y^{(g)}\}_{g=1}^{G} \sim \pi_\theta(\cdot \mid x)$
  for $g = 1, \ldots, G$ **do**
    Split reasoning and final answer separately into sentences $s_{g,1:m_g}$
    for $i = 1, \ldots, m_g$ **do**
      Extract the first valid entity pair from $s_{g,i}$ and query its count $c_{g,i}$
      Use $c_{g,i} = \perp$ if no valid pair or usable count is available
      $r^{\mathrm{c}}_{g,i} \leftarrow$ Eq. (2)
    end for
    Compute $R^{\mathrm{j}}(x, y^{(g)})$ and $R^{\mathrm{f}}(y^{(g)})$
    Build alignment σ_g and token returns $R_{g,t}$ via Eq. (3)
    $\bar{R}_g \leftarrow T_g^{-1} \sum_{t=1}^{T_g} R_{g,t}$ over valid tokens
  end for
  $\mu_x \leftarrow \mathrm{mean}_g(\bar{R}_g)$, $s_x \leftarrow \mathrm{std}_g(\bar{R}_g)$
  $A_{g,t} \leftarrow (R_{g,t} - \mu_x)/(s_x + 10^{-4})$
  Update LoRA with the clipped GRPO objective and KL to the frozen initial base model
end for

C.4 Qualitative Zero-Count Cases

Figure 7 shows three single-sentence errors penalized by zero corpus counts but accepted by GPT-4o-mini. Below, two complete Llama-3.1-8B-Instruct CorVer responses to NQ-Open questions have correct final answers despite flawed reasoning. Sentence IDs link each response to the extracted triplets, counts, rewards, and factual assessments.

Error type	Generated claim and indexed entity pair
A. Digit substitution	Ronkonkoma's ZIP code is **11777**. Gold: 11779. Indexed pair: (Ronkonkoma, 11777).
B. Entity substitution	**Jennifer Coolidge** appears in *Grown Ups*. Gold: Maria Bello. Indexed pair: (Jennifer Coolidge, Grown Ups).
C. Near-name substitution	Ella Fitzgerald's mother was **Height Fitzgerald**. Indexed pair: (Ella Fitzgerald, Height).

All three cases: count $= 0$ CorVer: reject ($r_i^c = -0.2$) GPT-4o-mini: accept

Figure 7: Three single-sentence errors. Zero corpus co-occurrence penalizes each fabricated entity association, while GPT-4o-mini accepts the claim.

Complete responses and sentence rewards. Case 1 contains a fabricated name and an incorrect detail outside the first triplet. Case 2 contains a fabricated category association. Both tables separate factual assessment from co-occurrence rewards.

Case 1. Philadelphia Flyers

Question: when was the last time philly won the stanley cup
Reference answer: 1975

Generated reasoning
[1]The Philadelphia Flyers, a professional ice hockey team based in Philadelphia, Pennsylvania, won the Stanley Cup in 1975. [2]The team defeated the Buffalo Sabres in the Stanley Cup Finals, winning the series 4–2 with a game 5 victory on May 19, 1975. [3]**Philipp Mehldau** was the Governor of Pennsylvania during this time.

Generated final answer
[4]The Philadelphia Flyers last won the Stanley Cup in 1975. [5]The Stanley Cup was last won by the Philadelphia Flyers in 1975.

Sentence rewards

ID	Extracted triplet	Count c_i	r_i^c	Assessment
1	(Philadelphia Flyers, won, Stanley Cup)	12,323	$+0.1$	Correct
2	(the team, defeated, Buffalo Sabres)	16,484	$+0.1$	Correct triplet*
3	(Philipp Mehldau, Governor of, Pennsylvania)	0	-0.2	Fabricated
4	(Philadelphia Flyers, last won, Stanley Cup)	12,323	$+0.1$	Correct
5	(Stanley Cup, last won by, Philadelphia Flyers)	12,323	$+0.1$	Correct

Reading the trace. Sentences 1, 4, and 5 correctly identify the Flyers' 1975 Cup, which remains their most recent. Sentence 3 invents a governor's name. The governor was Milton Shapp.

*__Coverage limitation.__ The Flyers defeated the Sabres 4–2, so sentence 2's extracted triplet is correct. Its game number and date are wrong: the deciding game was Game 6 on May 27, rather than Game 5 on May 19. These details fall outside the first extracted triplet. Checking secondary triplets could expose additional errors when those details are extracted. Appendix C.2 defines and compares these alternative rules.

Case 2. National Collegiate Athletic Association (NCAA) basketball championship

Question: who won the ncaa basketball championship in 1989
Reference answer: Michigan

Generated reasoning
[1]Michigan was led by Glen Rice, who was named the Most Outstanding Player. [2]Michigan won the championship by defeating Seton Hall 80–79. [3]The game took place at the King Dome in Seattle, Washington. [4]The 1989 NCAA Men's Division I Basketball Championship was the 61st edition of the NCAA **Field Hockey** Championships and was held on April 3, 1989.

Generated final answer
[5]The Michigan Wolverines won the NCAA basketball championship in 1989. [6]The 1989 NCAA Men's Division I Basketball Championship was held at the King Dome in Seattle, Washington.

Sentence rewards

ID	Extracted triplet	Count c_i	r_i^c	Assessment
1	(Michigan, led by, Glen Rice)	163	$+0.1$	Correct
2	(Michigan, defeated, Seton Hall)	389	$+0.1$	Correct
3	(the game, took place at, King Dome)	542	$+0.1$	Correct
4	(1989 NCAA Basketball Championship, edition of, NCAA Field Hockey Championships)	0	-0.2	Fabricated
5	(Michigan Wolverines, won, NCAA basketball championship)	3,500	$+0.1$	Correct
6	(1989 Championship, held at, King Dome)	10	0	Correct

Reading the trace. Glen Rice was the 1989 tournament's Most Outstanding Player. Michigan defeated Seton Hall 80–79 in overtime at the Seattle Kingdome. Sentence 4 fabricates a connection to field hockey and receives -0.2. Sentence 5 matches the reference answer. Sentence 6 is correct but receives a neutral reward because its count is 10.

D Results, Diagnostics, and Cost

D.1 Full Per-Model Results

Table 10 reports five-benchmark accuracy for all six models under the shared evaluation protocol (40,964 questions per model and variant). Raw uses the official base weights, and CorVer uses the final checkpoint after 100 optimizer steps.

Table 10: Full per-model evaluation accuracy (%) on the five closed-book QA datasets (TriviaQA validation, NQ-Open validation, PopQA test, SimpleQA, TruthfulQA), scored under the unified answer-required protocol on the full evaluation sets. CorVer uses a zero-count penalty of -0.2 and step-100 checkpoints.

Model	Variant	TriviaQA	NQ-Open	PopQA	SimpleQA	TruthfulQA
Llama-3.2-3B-Instruct	Raw	57.84	36.65	18.54	2.24	36.35
	CorVer	66.19	45.48	26.26	3.91	46.76
Qwen3-4B	Raw	52.32	24.82	17.73	2.96	45.29
	CorVer	57.57	31.86	22.16	4.23	50.31
Llama-3.1-8B-Instruct	Raw	74.67	44.93	31.97	5.69	46.27
	CorVer	81.50	53.07	36.69	6.70	49.45
Qwen3-8B	Raw	63.75	30.94	21.45	3.01	42.72
	CorVer	67.39	36.87	27.57	4.92	51.65
OLMo-2-13B-Instruct	Raw	68.62	33.60	26.47	2.20	41.37
	CorVer	81.27	49.00	38.68	5.89	53.73
Qwen3-14B	Raw	65.82	33.19	18.88	2.01	35.86
	CorVer	75.95	44.46	32.02	6.22	53.49

D.2 PopQA Gains by Popularity and Model Scale

Table 11 gives the complete gains underlying Figure 5, including the full-set gain for each model. The shared Q1–Q4 buckets contain 3,573, 3,561, 3,566, and 3,567 questions, respectively, totaling 14,267. The final row averages the gains equally across the six models.

Table 11: PopQA accuracy gains (CorVer minus Raw, pp) for six models grouped by scale. Q1 is the least popular quartile and Q4 the most popular. The four buckets contain 3,573, 3,561, 3,566, and 3,567 questions. All covers all 14,267 questions, and Mean gives equal weight to the six models.

Scale	Model	Q1	Q2	Q3	Q4	All
3–4B	Llama-3.2-3B	+7.33	+4.66	+6.98	+11.91	+7.72
	Qwen3-4B	+4.79	+3.59	+2.75	+6.59	+4.43
8B	Llama-3.1-8B	+1.54	+3.37	+6.70	+7.26	+4.72
	Qwen3-8B	+7.50	+4.18	+4.04	+8.77	+6.13
13–14B	OLMo-2-13B	+8.59	+7.08	+13.83	+19.37	+12.22
	Qwen3-14B	+12.37	+8.57	+11.44	+20.16	+13.14
Mean across six models		+7.02	+5.24	+7.62	+12.34	+8.06

D.3 Baseline Configuration Comparisons

FSPO and RLFH configuration comparisons start from the same official base weights with fresh LoRA adapters. They share the 6,621-question pool, rank/scaling $128/256$, seed 42, and 1,000-token response limit. Each of 100 steps samples 4 questions for Qwen3-4B and 3 for the other models. Final checkpoints use the five-benchmark scoring protocol in Appendix A.2.

D.3.1 FSPO Hyperparameter Comparison

Table 12 compares FSPO settings from Li & Ng (2025) (§6.1 and Appendix D) with our adapted configuration. The author-key runs use learning rate 4×10^{-7}, KL coefficient 0.001, rollout tempera-

ture 1.0, and 8 responses per question. The adapted configuration uses 2×10^{-6}, 0.04, 0.7, and 16, respectively.

These runs transfer the key hyperparameters to our shared setup. The original FSPO recipe instead uses batches of 8 questions, a 2,048-token response limit, and one epoch over a mixture of 2K QA examples and approximately 8K math problems.

The adapted configuration improves all 20 model–benchmark scores. Its macro accuracy is higher by 2.72, 0.36, 3.74, and 1.27 pp on Llama-3.2-3B, Qwen3-4B, Llama-3.1-8B, and Qwen3-8B, respectively. We retain this stronger FSPO configuration in the main comparison.

Table 12: FSPO accuracy (%) with author-reported key hyperparameters (Author key) and our adapted configuration (Main table), evaluated on the same full benchmark sets. Macro is the unweighted mean across benchmarks.

Configuration	TriviaQA	NQ-Open	PopQA	SimpleQA	TruthfulQA	Macro
Llama-3.2-3B						
Author key	57.96	36.81	18.55	2.50	37.82	30.73
Main table	59.45	40.25	22.11	4.30	41.13	**33.45**
Qwen3-4B						
Author key	53.23	26.01	17.87	3.05	44.68	28.97
Main table	53.55	26.79	18.14	3.37	44.80	**29.33**
Llama-3.1-8B						
Author key	74.94	44.40	32.23	5.69	46.14	40.68
Main table	79.81	51.02	36.77	6.89	47.61	**44.42**
Qwen3-8B						
Author key	64.34	31.94	21.90	3.56	44.92	33.33
Main table	64.77	32.24	22.77	3.91	49.33	**34.60**

D.3.2 RLFH Hyperparameter Comparison

We also rerun RLFH with author-reported key settings (Wen et al., 2025). Both configurations use learning rate 5×10^{-7}, 16 responses per question, and four PPO epochs per batch. Our adapted configuration uses KL coefficient 0.05 and rollout temperature 0.7, while the author-key runs use 0.01 and 1.0. The author-key runs use top-p 0.9, top-k 20, and presence penalty 1.5. These runs transfer key settings into our training setup rather than reproduce the original recipe in full.

The Qwen judge and reward rules follow the RLFH implementation in Appendix A.1.

The preferred configuration varies with the model (Table 13). The author-key settings improve macro accuracy by 6.51 pp on Llama-3.2-3B and 0.62 pp on Qwen3-4B. The adapted configuration remains stronger on Llama-3.1-8B and Qwen3-8B by 2.18 and 1.44 pp. The main comparison therefore uses the author-key runs for the two smaller models and the adapted runs for the two 8B models. Each selected row comes from one checkpoint across all five benchmarks.

Table 13: RLFH accuracy (%) with our adapted configuration and author-reported key settings. Bold macro values identify the complete rows selected for Table 1.

Configuration	TriviaQA	NQ-Open	PopQA	SimpleQA	TruthfulQA	Macro
Llama-3.2-3B						
Adapted	51.98	26.95	17.64	2.15	25.34	24.81
Author key	59.12	36.90	20.75	2.61	37.21	**31.32**
Qwen3-4B						
Adapted	50.18	25.01	16.22	1.78	33.17	25.27
Author key	50.68	24.60	15.99	1.60	36.60	**25.89**
Llama-3.1-8B						
Adapted	72.49	41.22	29.92	5.36	38.92	**37.58**
Author key	71.62	38.42	29.03	4.37	33.54	35.40
Qwen3-8B						
Adapted	62.77	30.39	20.52	2.91	42.35	**31.79**
Author key	62.20	28.86	19.82	2.94	37.94	30.35

D.3.3 FoRAG Batch Configuration Comparison

FoRAG releases model weights and datasets but not training code. The original paper specifies SFT settings, while its RL formulation leaves the numeric PPO learning rate, KL coefficient, and rollout temperature unspecified (Cai et al., 2024). Table 7 reports the settings used in our training setup.

Rollout allocation. We compare the initial and adjusted rollout allocations in Table 6 on all four main models. Both use LoRA rank/scaling $128/256$, learning rate 5×10^{-7}, KL coefficient 0.02, and four PPO epochs per batch. The main table uses adjusted runs for Llama-3.2-3B and Qwen3-8B and initial runs for the other models, all at rollout temperature 1.0. On Qwen3-8B, reducing generation from $4{,}800$ to 400 responses improves all five accuracies (Table 14).

Table 14: Qwen3-8B FoRAG accuracy under two rollout allocations. Accuracy is evaluated on all $40{,}964$ questions. Response counts describe training generation budgets. The adjusted run takes 5 h 42 min on H100 80GB.

Training responses	TriviaQA	NQ-Open	PopQA	SimpleQA	TruthfulQA	Macro
$4{,}800$ (original)	49.14	19.11	15.11	1.48	21.18	21.20
400 (adjusted)	60.86	28.42	19.81	2.73	41.00	30.57

D.3.4 KnowRL Training Duration Comparison

We compare step-100 and step-200 checkpoints on all four main models using the KnowRL configuration in Table 7. Each run continues for 100 additional optimizer steps with hardware, base-model revision, and training pool held fixed. Main accuracy and cost comparisons use step 100 and the timing scope in Appendix D.5. The duration diagnostics below focus on Qwen3-4B and Qwen3-8B at step 200, with complete training times of 21 h 54 min and 36 h 23 min, respectively.

Reward configuration. In our KnowRL configuration (Ren et al., 2025), the correctness term assigns $+2$ to correct answers, $+1$ to refusals, and -1 to incorrect answers. CorVer assigns $+2$ to correct answers and -1 to incorrect, refused, or empty answers (Appendix A.1.3). In answer-accuracy evaluation, refused and empty answers receive no credit and remain in the denominator for every method.

Accuracy and refusal. The original KnowRL study (Ren et al., 2025) examines 100–300 training steps and reports incorrect rate, refusal rate, precision on answered questions, and F1. Extending both Qwen runs from 100 to 200 optimizer steps did not improve macro accuracy. Qwen3-8B macro accuracy falls from 30.48% to 28.58% under the same evaluation protocol. For Qwen3-4B, macro accuracy falls from 20.80% to 19.07%, with refusal increasing on all five benchmarks.

The Qwen3-4B step-200 outputs contain 26,868 explicit refusals among 40,964 responses. The high refusal rate is consistent with KnowRL's abstention incentive and limits accuracy when refusals receive no credit. Table 2 of Ren et al. (2025) reports a related trade-off on Skywork-OR1-7B-Preview: the SimpleQA refusal rate rises from 21.33% to 37.67% while the incorrect rate falls from 76.33% to 60.33%, with precision on answered questions moving from 2.97 to 3.21 and F1 from 2.61 to 2.46. ChineseSimpleQA shows the same pattern (24.00% to 40.33% refusal, 67.00% to 52.33% incorrect). These results align with KnowRL's goal of avoiding incorrect answers through appropriate abstention. Its training-step analysis likewise reports refusal rates increasing and then plateauing over 100–300 steps (Ren et al., 2025, §4.5). Precision on answered questions includes only attempted answers, whereas our accuracy retains all questions and scores refusals as errors. The observed accuracy loss therefore reflects the cost of abstention under an answer-required metric, while reliability among attempted answers is a separate question.

D.4 Sensitivity to Answer Scoring

One-way substring matching. We rescore the Llama-3.2-3B-Instruct answers on all 35,821 TriviaQA, NQ-Open, and PopQA questions after removing the reverse-match branch. A match now requires the final answer to contain a reference answer or alias, with the remaining normalization and scoring rules unchanged. Table 15 shows that CorVer retains its lead. Its macro accuracy falls by 0.31 pp, compared with 1.48 for Raw and 1.24 for FSPO. The advantage over FSPO increases from 5.38 to 6.31 pp under the one-way rule.

Table 15: Full-set substring check for Llama-3.2-3B-Instruct. Reverse-only counts are pooled across the three datasets. Accuracy (%) is the unweighted mean of their individual accuracies.

Method	Reverse-only matches	Bidirectional	One-way
Raw	457	37.68	36.20
FSPO	432	40.60	39.36
CorVer	132	**45.98**	**45.67**

Semantic correctness on three QA datasets. We also score 100 questions from each of TriviaQA, NQ-Open, and PopQA using a remote GLM judge. Raw, FSPO, and CorVer are evaluated on the same 300 questions, giving 900 answers. The same judge and grading rules apply to all answers. Table 16 compares the original substring scores with semantic correctness on this same sample. CorVer leads under both measures, reaching 54.33% semantic correctness versus 47.33% for FSPO and 44.00% for Raw.

Table 16: Paired answer-scoring comparison (%) for Llama-3.2-3B-Instruct on 300 questions, with 100 from each of three QA datasets.

Method	Original substring	Semantic correctness
Raw	42.00	44.00
FSPO	47.00	47.33
CorVer	**52.33**	**54.33**

Truthfulness and informativeness on TruthfulQA. Using the same GLM judge, we assess the three methods on a shared sample of 100 TruthfulQA questions. We report the proportion of answers judged truthful and the proportion judged both truthful and informative, each over all 100 questions. CorVer reaches 55% truthfulness and 54% truthful-and-informative answers (Table 17). Its advantage on this sample therefore extends to semantic judgments beyond the main-table ROUGE rule. These sample-based checks supplement the full-set lexical scores.

D.5 Accuracy and Training Cost on H100

Measurement and comparison scope. Figure 2(b) and Table 18 pair accuracy and complete training time at step 100 for the selected Qwen3-4B and Qwen3-8B configurations. Each policy trains

Table 17: GLM semantic evaluation (%) of Llama-3.2-3B-Instruct on the same 100 TruthfulQA questions across methods. The second metric requires both truthfulness and informativeness.

Method	Truthful	Truthful and informative
Raw	44	34
FSPO	46	40
CorVer	**55**	**54**

on one H100 80GB. Timing covers initialization, rollout generation, reward computation, policy updates, and final saving, excluding job queueing and post-training evaluation. RLFH and KnowRL use remote judges: their inference, API communication, and response waiting contribute to elapsed time, while remote computation occurs outside the local H100. FSPO and FoRAG score locally. Raw provides an accuracy reference.

The selected RLFH runs use author-key settings on Qwen3-4B and adapted settings on Qwen3-8B. FoRAG uses the initial Qwen3-4B allocation and adjusted Qwen3-8B allocation (Appendix D.3). Workloads differ in generation budgets and update schedules: PPO reuses responses for four epochs, and adjusted Qwen3-8B FoRAG generates 400 responses versus CorVer's 4,800. Time ratios therefore compare complete training configurations rather than isolated verifier speed at matched workload.

Additional CorVer timings. Under the same timing scope, 100-step CorVer training takes 1 h 50 min on Qwen3-14B and 2 h 17 min on OLMo-2-13B-Instruct. These models are excluded from the two-model averages.

Aggregation. Accuracy averages the five benchmarks equally, and accuracy and time averages weight the two models equally. Each time ratio divides a baseline's mean complete-run duration by CorVer's. Calculations use unrounded values, while the table displays h:mm rounded to the nearest minute.

Table 18: Accuracy and training cost. All timed runs use one H100 80GB. All methods are compared at step 100, using the same accuracy evaluations as Table 1. Macro is five-benchmark macro accuracy (%). Time is training duration in h:mm (rounded to the nearest minute), excluding queueing and evaluation. Mean weights the two models equally. The last column gives mean time relative to CorVer.

		Qwen3-4B		Qwen3-8B		Mean		Time ratio
Method	Steps	Macro	Time	Macro	Time	Macro	Time	/ CorVer
Raw	—	28.62	—	32.37	—	30.50	—	—
CorVer	**100**	**33.22**	**2:01**	**37.68**	**1:36**	**35.45**	**1:49**	1.00×
FSPO	100	29.33	10:58	34.60	9:04	31.97	10:01	5.52×
RLFH	100	25.89	15:30	31.79	22:15	28.84	18:53	10.41×
FoRAG	100	26.96	22:28	30.57	5:42	28.76	14:05	7.76×
KnowRL	100	20.80	12:58	30.48	18:51	25.64	15:55	8.77×