Beyond RLVR: Verifiable Reward Design for Open-Ended Generation

Anonymous ACL submission

Abstract

Open-ended generation has no natural reward: there is no gold answer to verify against, and querying an LLM judge at every step is costly and easy to game. We ask whether the training signal can instead come from checkable properties of grounded generation itself. EGVR (Entailment-Grounded Verification Rewards) scores three such properties: source-grounded entailment (R_{ent}), cross-rollout self-consistency (R_{con}), and causal faithfulness to input evidence (R_{caus}). We train Qwen2.5-7B-Instruct with GRPO on grounded RAGTruth prompt-source pairs and evaluate whether reward optimization transfers to held-out quality metrics. The optimized EGVR reward increases substantially during training, while external quality remains stable rather than collapsing. The final model (step=1000) improves WildBench score over the baseline, shows positive gains on ExpertQA factuality metrics, and remains near parity on Arena-Hard. Paired ExpertQA comparisons are nearly balanced, indicating that the gains are modest but consistent with improved factual grounding. Our results suggest that property-based rewards can provide useful training signals for open-ended grounded generation when anchored to external evidence and evaluated with held-out benchmarks. More broadly, EGVR offers a practical reward design and evaluation protocol for studying open-ended RL beyond exact-answer domains, reducing reliance on gold responses and generative judges while preserving downstream generation quality.

1 Introduction

Recent advances in reinforcement learning have transformed large language model post-training. Reinforcement Learning from Human Feedback (RLHF) aligns model outputs with human preferences through learned reward models (Ouyang et al., 2022; Bai et al., 2022), while Reinforcement Learning with Verifiable Rewards (RLVR) replaces learned rewards with deterministic checkers, achieving strong results in mathematics and code (Guo et al., 2025; Shao et al., 2024; Kazemnejad et al., 2024; Lambert et al., 2024).

Despite these successes, both paradigms face fundamental limitations in open-ended settings. RLHF requires an LLM judge at every gradient step, which is costly, unreliable, and exposed to judge artifacts such as length and verbosity bias, self-preference, and outright reward hacking (Wang et al., 2024; Raina et al., 2024; Skalse et al., 2022; Pan et al., 2022). RLVR, while avoiding judges, demands domain-specific verifiers: no deterministic verifier can certify a summary or long-form answer the way a unit test certifies code. These constraints limit both paradigms to narrow domains and complicate deployment where ground truth might be unavailable. This motivates a central question: *Can property-based verification signals, deterministic checks on logical properties of outputs, substitute for a judge in open-ended RL post-training?*

In this paper, we introduce and study such signals under a framework we call EGVR (Entailment-Grounded Verification Rewards). Rather than scoring responses with an LLM judge, EGVR checks three properties: whether claims are supported by a source document (R_{ent}), whether the model makes consistent commitments across repeated samples (R_{con}), and whether responses actually depend on the prompt via counterfactual sensitivity (R_{caus}). All three are deterministic and run entirely from local model weights.

Under EGVR, we train Qwen2.5-7B-Instruct via GRPO with LoRA, evaluating across Arena-Hard creative writing (Li et al., 2024), ExpertQA factual QA (Malaviya et al., 2024), and WildBench instruction following (open-ended questions subset) (Lin et al., 2024). These benchmarks are selected to evaluate complementary aspects of reference-free grounded GRPO with EGVR. ExpertQA most directly reflects the core objective, emphasizing

domain-specific long-form factual QA with source support, attribution, and factual consistency. WildBench assesses transfer to a heterogeneous distribution of realistic open-ended user tasks, including context-rich and constraint-driven prompts. Arena-Hard serves as a general instruction-following regression benchmark, testing whether improved grounding preserves broad chat capability.

Our contributions are:

1. We introduce EGVR, a reference-free and LLM-judge-free composite reward for open-ended generation, combining rollout consistency, source-grounded entailment, and causal sensitivity to input evidence. This design reduces reliance on gold responses or expensive preference judges while remaining applicable to grounded long-form generation.
2. We propose a reward design and evaluation protocol aimed at mitigating reward hacking. EGVR decomposes reward into interpretable verifier-based signals and is evaluated against held-out external benchmarks, allowing us to check whether reward improvements correspond to stable downstream quality rather than optimization artifacts.
3. We show that grounded GRPO with EGVR yields modest improvements in open-ended factuality and transfer while largely preserving general generation quality. In particular, ExpertQA and WildBench show small positive gains, while Arena-Hard remains near parity, suggesting that improved grounding and evidence sensitivity need not substantially degrade broader instruction-following or creative-writing performance.

2 Related Work

Open-ended RL with generative judges and rubrics. Preference-based post-training methods such as RLHF and direct preference optimization train language models from human or model preferences (Ouyang et al., 2022; Bai et al., 2022; Rafailov et al., 2023; Ethayarajh et al., 2024). For open-ended generation, where exact answers are unavailable, recent work increasingly relies on LLM judges or rubric-based evaluators to provide reward signals during training (Zheng et al., 2023; Dubois et al., 2024; Gunjal et al., 2025; Shen et al., 2026), including self-rewarding setups where the model judges its own outputs (Yuan et al., 2024). These approaches are flexible but inherit known judge weaknesses, including verbosity bias, position sensitivity, and self-preference (Singhal et al., 2023; Wang et al., 2024; Panickssery et al., 2024). Our work studies a complementary direction: using non-generative, property-based verification signals as rewards for open-ended grounded generation, where the reward is anchored to external evidence rather than learned preferences.

Factuality, entailment, and consistency as verification signals. A line of work evaluates long-form generation by checking whether generated claims are supported by external evidence. FActScore decomposes generations into atomic facts and measures the fraction supported by a knowledge source (Min et al., 2023). NLI-based methods such as SummaC and WiCE study entailment between generated claims and evidence, highlighting the importance of claim decomposition, evidence retrieval, and fine-grained support judgments (Laban et al., 2022; Kamoi et al., 2023). Complementarily, SelfCheckGPT estimates hallucination risk through cross-sample agreement without relying on an external evidence source (Manakul et al., 2023). RLEF is the closest training-time precedent, using reference-free entailment rewards for factual summarization (Roit et al., 2023). EGVR extends this idea to broader grounded generation with a composite reward combining entailment, rollout consistency, and counterfactual sensitivity.

Verifiable rewards beyond exact-answer domains. Reinforcement learning with verifiable rewards has achieved strong results in domains such as mathematics and code, where correctness can be checked with exact or programmatic verifiers (Shao et al., 2024; Guo et al., 2025; Lambert et al., 2024; Lightman et al., 2024). Extending RLVR to open-ended generation is harder because there is no single correct answer and no ground-truth reference for comparison. Recent work explores self-consistency, majority voting, self-certainty, or generative reward models as alternatives (Zuo et al., 2025; Jia et al., 2025; Zhao et al., 2026). Our setting differs in that the reward is neither an exact-answer checker nor a generative judge; it is a composite of evidence-grounded verification signals designed for long-form source-grounded generation, combining external grounding with deterministic logical properties.

Reward hacking and held-out evaluation. Optimizing proxy rewards can improve the measured reward while failing to improve the intended objective, a phenomenon commonly discussed as Goodhart's Law or reward hacking (Goodhart, 1984; Manheim and Garrabrant, 2019; Gao et al., 2023). This risk is especially important for open-ended RL because factuality, consistency, and judge scores are only proxies for downstream response quality. We therefore evaluate reward optimization on held-out benchmarks rather than treating reward increase itself as success. Our results suggest that anchoring rewards to external evidence can improve the optimized reward while largely preserving downstream quality.

3 Method

We frame post-training as a reward design problem for grounded open-ended generation tasks (e.g., factual QA, summarization) where external evidence is available to anchor the training signal, but no gold response serves as a reward target. The policy π_θ is prompted with both task x and source evidence S and samples a group of K responses $Y = \{y_1, \ldots, y_K\}$. Each response receives an Entailment-Grounded Verification Reward $R_{\text{EGVR}}(y_i, S, Y)$, computed as

$$\lambda R_{\text{ent}}(y_i, S) + \beta R_{\text{con}}(y_i, Y) + \gamma R_{\text{caus}}(y_i, S). \tag{1}$$

where λ, β, and γ weight source entailment, self-consistency, and causal faithfulness, respectively. The reward combines three complementary signals: support by the provided evidence, consistency across independent samples, and sensitivity to factual changes in the evidence.

All entailment probabilities come from a frozen DeBERTa-v3-large NLI model (He et al., 2021), denoted $V(a, b)$. No generative judge scores model outputs during training. Counterfactual sources are built once as offline preprocessing (spaCy (Honnibal et al., 2020) where entity swap succeeds, with an LLM fallback otherwise), not as a per-step reward call.[1]

3.1 Source Entailment (R_{ent})

The source entailment reward measures whether the claims in a response are supported by the evidence. We split the source into overlapping segments $\text{Seg}(S) = \{S_1, \ldots, S_M\}$ and the response into sentence-level claims $\mathcal{C}(y_i) = \{c_1, \ldots, c_L\}$, then score each claim against its best-matching source segment:

$$R_{\text{ent}}(y_i, S) = \frac{1}{|\mathcal{C}(y_i)|} \sum_c \max_{S_m} V(S_m, c). \tag{2}$$

This term directly rewards grounding in the provided source without requiring a reference answer.

3.2 Self-Consistency (R_{con})

The self-consistency reward measures whether a response's central commitments are stable across repeated sampling. For each response, we extract an opening segment o_i from the first few sentences and compare it with the corresponding segments from the other rollouts:

$$R_{\text{con}}(y_i, Y) = \frac{1}{K-1} \sum_{j \neq i} V(o_i, o_j). \tag{3}$$

This provides a lightweight semantic agreement signal while avoiding the overly strict requirement that long open-ended generations mutually entail each other in full.

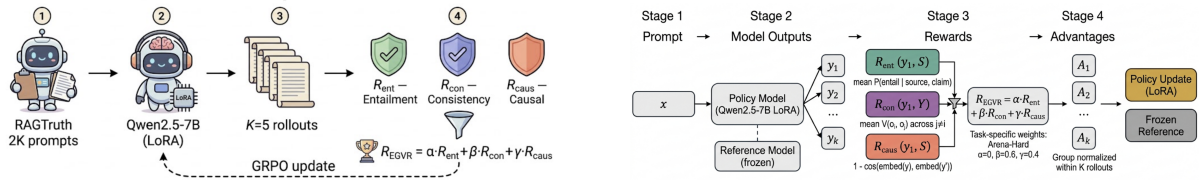
3.3 Causal Faithfulness (R_{caus})

Entailment and consistency can both be gamed by writing shorter, vaguer responses that lack specificity. R_{caus} guards against this by checking whether the response actually depends on the specific facts in the prompt, not just the general topic. We swap one verifiable element in the source (a number, date, or named entity via spaCy (Honnibal et al., 2020)) to produce a counterfactual S'. This mirrors counterfactually-augmented data, where minimal edits that flip a label isolate the feature a model should depend on (Kaushik et al., 2020). A response earns causal credit when it uses the original fact, does not already state the swapped value, and the paired counterfactual response adapts accordingly:

$$R_{\text{caus}}(y_i, S) = \text{clip}\big(u_i(1-b_i)\max(0,\ s_i - a_i),\ 0,\ 1\big), \tag{4}$$

where $u_i = V(S, y_i)$ is the NLI support for the original fact, $b_i = V(S', y_i)$ for the swapped fact, and $s_i = V(S', y_i')$, $a_i = V(S, y_i')$ are the analogous scores for the counterfactual response y_i'. Faithful adaptation means $s_i > a_i$ (the counterfactual response supports the swapped fact more than the original), so the term $\max(0, s_i - a_i)$ is positive precisely when the model correctly adapts.

[1] The model string for the judge used in evaluation is `gpt-4.1-2025-04-14`.



(a) EGVR reward components: source entailment, self-consistency, and causal faithfulness combined into R_{EGVR}.

(b) Training pipeline: per-rollout rewards are group-normalized across K rollouts and used to update the policy via GRPO with LoRA.

Figure 1: EGVR overview (left: reward components; right: training pipeline).

3.4 GRPO Optimization

Rewards are normalized within each rollout group before computing policy gradients:

$$A_i = \frac{R_{\text{EGVR}}(y_i, S, Y) - \mu_Y}{\sigma_Y + \epsilon}, \tag{5}$$

where μ_Y and σ_Y are the group mean and standard deviation. The policy is updated with a clipped objective and a KL penalty to the reference model:

$$\mathcal{L} = -\min(\rho_i A_i, \bar{\rho}_i A_i) + \beta_{\text{KL}} \widehat{\text{KL}}(\pi_\theta, \pi_{\text{ref}}). \tag{6}$$

where $\bar{\rho}_i$ is ρ_i clipped to $[1 - \epsilon_{\text{clip}}, 1 + \epsilon_{\text{clip}}]$, $\rho_i = \pi_\theta(y_i|x)/\pi_{\text{ref}}(y_i|x)$ is the importance ratio, π_{ref} is the frozen initial policy, and $\widehat{\text{KL}}(\pi_\theta, \pi_{\text{ref}}) = \log \pi_\theta(y_i|x) - \log \pi_{\text{ref}}(y_i|x)$ is the per-token KL estimator following Shao et al. (2024).

We initialize from Qwen2.5-7B-Instruct (Yang et al., 2024) and train LoRA adapters (Hu et al., 2022).

4 Experimental Setup

4.1 Training Data

We train on RAGTruth (Niu et al., 2024), a corpus of retrieval-augmented generation tasks with prompts and source documents. For each example, we use only the prompt and associated evidence, discarding the dataset-provided model responses as training targets. The responses are used only to recover official source identifiers and split metadata. Since multiple responses may share the same source, we de-duplicate by `source_id`.

After filtering and de-duplication, the training run contains 2,395 grounded prompt-source examples, with 266 examples held out for internal reward diagnostics. This holdout is used only for reward sanity checks, not for final benchmark reporting or model selection. Because RAGTruth is not part of our final evaluation suite, we use both its train and test source IDs as the prompt-source pool.

Each input contains the source evidence followed by the user task. This setup isolates the intended setting: the model must answer using provided evidence, without any gold answer supplied during optimization.

4.2 Model and Optimization

Our base model is Qwen2.5-7B-Instruct (Yang et al., 2024). We fine-tune it with LoRA adapters applied to the attention projection matrices `q_proj`, `k_proj`, `v_proj`, and `o_proj`, using rank 16, LoRA alpha 32, dropout 0.05, and bf16 training. This yields roughly 10.1M trainable parameters, or 0.13% of the full model.

We optimize with GRPO. For each prompt, the policy samples a group of five responses. Rewards are computed independently for each response, normalized within the group, and used as advantages in a clipped policy objective. We use clip coefficient 0.2 and a KL penalty of 0.01 against the frozen base policy. Training uses AdamW with learning rate 1×10^{-6}, cosine decay, 100 warmup steps, gradient clipping at 1.0, batch size 4 prompts, and 1,000 total update steps. Checkpoints are saved every 200 steps.

Rollouts are sampled with temperature 1.0, top-p 0.9, and a maximum of 512 new tokens. Prompts are truncated to 1,024 tokens during generation, and prompt-plus-completion sequences are truncated to 1,536 tokens for the policy loss.

4.3 EGVR Reward

The training reward combines three verifier signals:

$$R_{\text{EGVR}} = R_{\text{con}} + R_{\text{ent}} + 0.5\, R_{\text{caus}}.$$

We weight R_{con} and R_{ent} equally because they are dense signals available on every rollout; R_{caus} is discounted by 0.5 because entity-swap counterfactuals succeed on only a subset of prompts, giving it higher per-step variance.

All three terms are computed without reference answers. The verifier is a frozen `cross-encoder/nli-deberta-v3-large` model and is never fine-tuned.

4.4 Evaluation Benchmarks

We evaluate on three held-out benchmarks that probe complementary aspects of open-ended generation: factual grounding, broad real-user transfer, and general instruction-following quality.

ExpertQA. We evaluate long-form domain-specific factual QA on ExpertQA (Malaviya et al., 2024), using the `cmalaviya/expertqa` dataset with the `main` configuration. Because ExpertQA emphasizes factuality in expert-oriented responses, it is closest to our target setting; we therefore report FAct-hard as the primary factuality metric.

Each example is converted into a prompt-source pair: the prompt asks the model to answer thoroughly and factually, while the source is formed by concatenating the evidence strings from the reference answer object `rr_gs_gpt4`. We remove examples with empty questions, missing reference answers, or no usable evidence. For inference, we generate one rollout per question using the model's chat template without a separate system prompt. The user message contains the question and evidence. Generation uses temperature 0.8, top-p 0.9, a 512-token output limit, and a 3,072-token prompt truncation limit.

We score outputs with the same frozen NLI verifier used by EGVR, `cross-encoder/nli-deberta-v3-large`. Answers are sentence-split, retaining sentences with at least three words. Evidence is chunked into 180-word sliding windows with 40-word overlap, capped at 8 chunks. For each answer sentence, we take the maximum entailment probability over chunks. FAct-soft is the mean entailment probability across sentences, and FAct-hard is the fraction of sentences with entailment score at least 0.5. We report both metrics averaged across examples.

WildBench. We evaluate broad open-ended transfer on WildBench v2 (Lin et al., 2024) using the official AllenAI implementation and evaluator. We apply only a minimal local patch to expose open-ended dataset views; the official scoring prompt, OpenAI batch-evaluation flow, and WB-Score computation remain unchanged.

Our primary evaluation uses the `open_ended` subset from the `allenai/WildBench` v2 test split. Following WildBench's first-tier `primary_tag`, we include Creative Writing, Brainstorming, Planning, Editing, Advice seeking, and Role playing, yielding 419 examples out of 1,024 total test examples.

We choose this subset because our method targets reference-free open-ended generation, where quality depends on helpfulness, coherence, specificity, instruction following, and context use rather than exact-answer correctness. We exclude Coding & Debugging, Math, Data Analysis, and Reasoning, which often require execution-based, symbolic, or task-specific evaluation. We also exclude Information seeking because it overlaps with factual QA, which is measured more directly by ExpertQA.

For scoring, we use the official WildBench GPT-4o judge, `gpt-4o-2024-05-13`. The evaluator assigns a raw 1–10 score, converted to WB-Score as $2(s-5)$. LoRA checkpoints are merged into standalone Hugging Face model directories before inference. We use the official Hugging Face engine with the model's native chat template, bf16 weights, maximum generation length 4096, temperature 0.0, and top-p 1.0. Unlike GRPO training, WildBench uses one official response per prompt rather than multiple rollout groups.

Arena-Hard. We evaluate the Arena-Hard-v2.0 creative-writing subset (250 prompts) as a regression check for broad instruction-following quality. Since EGVR targets grounded generation rather than unconstrained creative writing, we use this benchmark primarily as a guardrail: factuality improvements should not cause large declines in general response quality.

We compare one checkpoint rollout against the base Qwen2.5-7B-Instruct model using pairwise LLM-as-judge evaluation. GPT-4o-mini is used as the primary judge, with GPT-4.1 verification on the same prompt pairs. Response order is randomized with a 50% swap rate to control position bias, and API calls use exponential backoff with up to three retries. We report win rate against the base model.

Unless a benchmark evaluator specifies otherwise, evaluation generation uses the same decoding setup as training: five sampled rollouts, temperature 1.0, top-p 0.9, and 512 maximum new tokens. We evaluate the base model and checkpoints at steps 200, 400, 600, 800, and 1000 where available.

5 Results

We evaluate grounded GRPO with EGVR on three held-out open-ended benchmarks: ExpertQA, WildBench open-ended, and Arena-Hard. Our evaluation is designed to answer two questions: whether EGVR improves downstream model quality, and whether reward optimization remains aligned with external evaluation rather than producing reward-hacking artifacts.

5.1 Downstream Model Quality

5.1.1 Best-Checkpoint External Evaluation

Benchmark	Metric	Base	EGVR	Δ	Rel. %
ExpertQA	FAct-soft	.619	.635	+.015	+2.48
	FAct-hard	.743	.760	+.017	+2.25
WildBench	WB-score	4.945	5.041	+.096	+1.93
	Raw score	7.473	7.520	+.048	+0.64
Arena-Hard	Win (%)	50.0	58.4	+8.4	+16.80

Table 1: Best-checkpoint results across benchmarks, showing improvements over the base model.

Table 1 shows that EGVR improves best-checkpoint performance across all three benchmarks. ExpertQA gains indicate better NLI-measured grounding, though these scores use the same DeBERTa verifier as the reward and may partly reflect reward alignment. WildBench improves under both raw and normalized scores, suggesting some transfer beyond the training setting. Arena-Hard also improves, indicating that grounding-oriented optimization does not degrade broad instruction-following quality.

Overall, EGVR yields consistent external gains while preserving broader generation quality. The reward uses no gold responses or preference annotations, yet produces measurable improvements on held-out evaluations.

5.1.2 Checkpoint-Level Behavior

Step	EQA-S	EQA-H	WB-Raw	WB	Arena
0	.619	.743	7.473	4.945	50.0
200	.627	.760	7.480	4.959	51.2
400	.623	.752	7.506	5.012	54.0
600	.619	.743	7.485	4.969	58.4
800	.635	.759	7.520	5.041	49.2
1000	.625	.751	7.511	5.022	50.8

Table 2: Checkpoint-level external evaluation results.

Table 2 shows that EGVR gains are stable but not monotonic. ExpertQA and WildBench remain near or above the baseline across most checkpoints, suggesting that the reward provides a useful training signal rather than producing a single isolated improvement. However, the best external performance does not always occur at the final checkpoint: ExpertQA and WildBench peak around step 800, while Arena-Hard is highest at step 600. This indicates that external validation remains necessary for checkpoint selection in open-ended RL.

5.2 Reward-Hacking Robustness

5.2.1 Goodhart Analysis

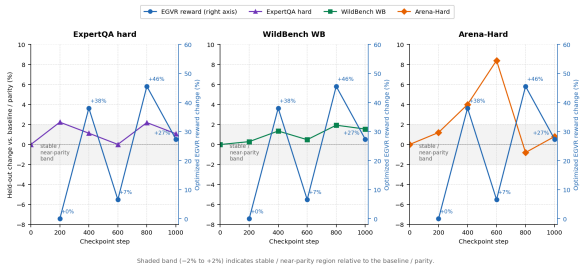


Figure 2: Goodhart analysis across checkpoints.

Figure 2 shows that increasing R_{EGVR} does not coincide with systematic held-out degradation. ExpertQA and WildBench stay near or above baseline, while Arena-Hard fluctuates without progressive decline. This suggests that EGVR avoids the strongest reward-hacking pattern, where reward rises while external quality collapses.

5.3 Training Stability and Collapse Diagnostics

Figure 3 shows that response length, refusal rate, and generation-ratio diagnostics all remain stable across 1,000 GRPO steps. The model does not shorten responses to exploit the reward, does not learn to refuse difficult prompts, and stays close to the base policy's output distribution. These diagnostics rule out the most obvious degenerate patterns that could trivially inflate reward without improving quality.

6 Analysis

6.1 Training Reward Dynamics

As shown in Figure 4, the composite R_{EGVR} reward exhibits a modest upward trend over the course of GRPO training. The raw per-step reward values are noisy, with several temporary drops, but the smoothed trajectory and reference line indicate that the optimized reward generally increases during

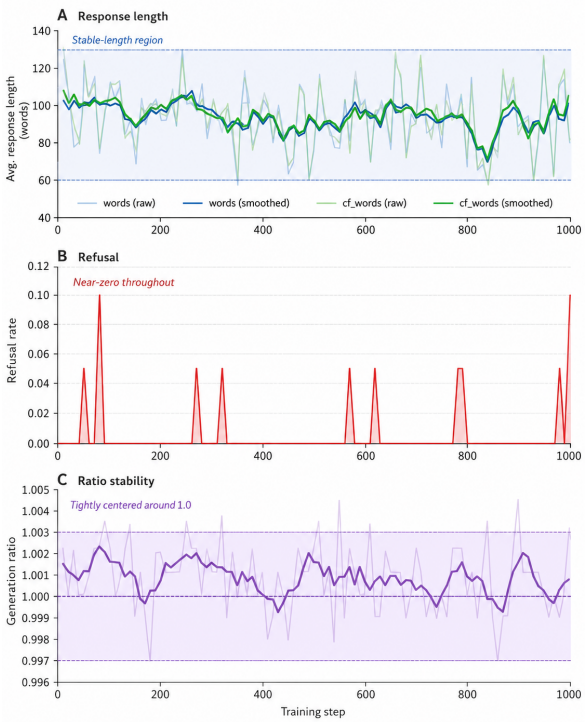


Figure 3: Training stability diagnostics during GRPO.

training. This trend suggests that GRPO is able to improve the intended training signal rather than leaving the verifier reward unchanged.

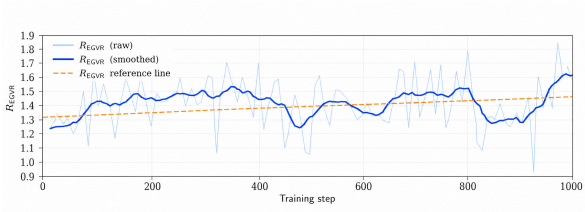


Figure 4: Training trajectory of the composite R_{EGVR} reward under GRPO.

6.2 Ablation Analysis

Table 3 isolates the stability role of causal faithfulness. R_{con} alone is initially near parity but collapses from 51.6% at step 50 to 22.4% at step 100, showing that self-consistency alone can be exploited by degenerate or underspecified responses. Adding R_{caus} avoids this collapse and reaches 57.2% win rate at step 200, suggesting that causal sensitivity makes the agreement signal harder to game by requiring responses to depend on input-specific factual changes. This supports the composite design of EGVR: R_{con} supplies a dense agreement signal, R_{caus} checks factual dependence, and R_{ent} anchors the reward to external evidence in grounded generation.

Config	Step	GPT-4.1 win rate	Interpretation
R_{con} only	50	51.6	initially stable
R_{con} only	100	22.4	collapse
R_{con}+R_{caus}	200	**57.2**	avoids collapse

Table 3: Reward component ablation on Arena-Hard. R_{con} alone collapses under continued optimization, while adding R_{caus} avoids this failure mode and improves over the untuned baseline.

We further test whether the stability pattern scales beyond the original Qwen2.5-7B setting. On Qwen3-8B with UltraFeedback, R_{con}+R_{caus} remains stable over 1,000 GRPO steps: the composite reward increases from 0.084 to 0.151, with a 2.4% smoothed early-to-late gain and no collapse. Together with the component ablation above, this suggests that causal sensitivity is a scalable stabilizer for consistency-based rewards. Overall, the ablations support the EGVR design: open-ended reward construction benefits from combining source grounding, rollout consistency, and causal sensitivity, with each component addressing a different failure mode.

7 Discussion

EGVR shows that source-grounded open-ended generation can be optimized with property-based verifier rewards rather than gold answers or LLM-judge rewards. Although open-ended tasks lack a single correct answer, grounded generation still exposes checkable properties: source support, cross-rollout consistency, and sensitivity to evidence changes.

A key design principle is that verifier rewards should be externally anchored and independently validated. Source evidence limits the model's ability to game the reward, while held-out benchmarks test whether reward gains transfer to downstream quality. Thus, EGVR is best viewed not as a complete replacement for judge-based evaluation, but as a practical training signal and evaluation protocol for open-ended RL.

8 Conclusion

We studied whether open-ended grounded generation can be optimized without gold responses or LLM-judge rewards. To this end, we introduced EGVR, a reference-free and LLM-judge-free reward framework that combines source-grounded

entailment, cross-rollout consistency, and causal sensitivity to input evidence. Unlike exact-answer RLVR, EGVR does not require a single correct target answer; unlike judge-based open-ended RL, it does not call a generative evaluator during training. Instead, it uses checkable properties of grounded generation as the optimization signal.

Our experiments show that EGVR provides a useful training signal. The optimized reward increases during GRPO training, and held-out evaluation shows consistent positive gains on factual and open-ended benchmarks while largely preserving broader instruction-following quality. ExpertQA and WildBench improve over the base model, while Arena-Hard remains near parity. Checkpoint-level results are not monotonic, confirming that external validation remains necessary for model selection. Diagnostic analyses show that the reward increase is not explained by the most obvious collapse modes: responses do not shorten toward empty outputs, refusals remain rare, and generation-ratio stability remains close to the base policy.

More broadly, our findings suggest a practical design principle for RL in non-verifiable domains: anchor rewards to information outside the policy's control, then validate against independent held-out tasks. Source evidence provides one such anchor, because the model cannot change the documents against which its claims are checked. Self-consistency and causal sensitivity provide additional signals. EGVR should be viewed not as a replacement for broad evaluation, but as a training-time alternative to expensive judge-based rewards. We hope this work encourages further study of property-based reward design for open-ended RL.

Limitations

EGVR depends on learned verifier models. In this work, factual support is estimated mainly with a frozen DeBERTa NLI verifier, which makes the reward sensitive to that verifier's calibration and failure modes. This is especially important for ExpertQA: our automatic factuality scores use the same verifier family as the training reward, so improvements may partly reflect alignment with the verifier rather than genuine factuality gains. NLI scoring is also limited for numerical reasoning, negation, coreference, domain-specific terminology, and claims whose support is distributed across multiple evidence spans. More robust alignment metrics, such as AlignScore, or evaluations with multiple independent verifiers would strengthen future work.

EGVR is designed for grounded generation with explicit source evidence. It does not directly apply to open-ended tasks without reliable evidence, and it optimizes only verifiable aspects of quality. Other important dimensions, such as coverage, pedagogy, style, creativity, and user preference, are not directly rewarded. The reward components are also approximate: sentence-level entailment misses discourse-level commitments, opening-segment consistency does not guarantee consistency across long responses, and local counterfactual edits test only a limited class of factual dependencies.

Our empirical results are bounded by the training setup. We train LoRA adapters for one 7B instruction model with five rollouts per prompt and 1,000 GRPO updates. Larger models, full-parameter training, other RL algorithms, longer training runs, and different evidence distributions remain untested. Although EGVR avoids per-step generative LLM judges, it still requires many verifier calls over claims, chunks, and rollouts, so its cost is lower than judge-based RL but not negligible.

Finally, evaluation of open-ended generation remains approximate. WildBench and Arena-Hard rely on LLM judges, whose scores can vary with judge model, prompting protocol, response order, and stylistic preferences. ExpertQA is more factuality-oriented, but automatic scoring cannot replace expert human assessment. We therefore interpret our results as evidence that EGVR improves verifier-aligned grounded behavior, not as proof of overall response quality.

Ethical Considerations

This work studies reward design for source-grounded open-ended generation. The proposed method is intended to improve factual grounding and reduce unsupported generation, but it does not eliminate hallucinations or guarantee truthfulness. Because verifier-based rewards can themselves be imperfect, model outputs should not be treated as reliable in high-stakes settings without additional human or domain-expert review.

Our experiments use existing benchmark datasets and automatic evaluators. We do not collect new human-subject data. When using model-based evaluators, we recognize that the evaluator

may encode biases, may be sensitive to prompt formatting, and may fail to capture all dimensions of response quality. We therefore report both downstream evaluation results and diagnostic analyses rather than relying on reward values alone.

AI Assistance Disclosure. AI assistance tools, including ChatGPT, Codex, Claude, Claude Code, and Gemini, were used for language polishing, coding assistance, debugging support, and analysis/visualization scripting.

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, and 1 others. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2024. Alpacafarm: A simulation framework for methods that learn from human feedback. 36.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Jean-Baptiste Dujardin, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. pages 10011–10040.

Charles AE Goodhart. 1984. Problems of monetary management: The uk experience. *Monetary Theory and Practice*, pages 91–121.

Tushar Gunjal and 1 others. 2025. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*.

Daya Guo, Dejian Yang, He Zhang, Junxiao Song, Runxin Zhang, Ruyi Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Pengcheng He, Jianzhuo Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Yufei Jia and 1 others. 2025. Writing-zero: Bridging non-verifiable tasks and verifiable rewards. *arXiv preprint arXiv:2506.00103*.

Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023. Wice: Real-world entailment for claims in wikipedia. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 5109–5128.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Amirhossein Kazemnejad, Milad Aghajohari, Pascal Poupart, and Alessandro Sordoni. 2024. Vineppo: Unlocking rl potential for llm reasoning through refined credit assignment. *arXiv preprint arXiv:2410.01679*.

Philippe Laban, Tobias Schnabel, Paul Bennett, and Marti A Hearst. 2022. Summac: Re-visiting nli-based models for inconsistency detection in summarization. volume 10, pages 163–177.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, and 1 others. 2024. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024. From live data to high quality benchmarks: The arena-hard pipeline. *arXiv preprint arXiv:2406.11939*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *International Conference on Learning Representations*.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Megan Sclar, Khyathi Chandu, Chandrasekhar Bhagavatula, and Yejin Choi. 2024. Wildbench: Benchmarking llms with challenging tasks from real users in the wild. *arXiv preprint arXiv:2406.04770*.

Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2024. Expertqa: Expert-curated questions and attributed answers. *arXiv preprint arXiv:2309.07852*.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

David Manheim and Scott Garrabrant. 2019. Categorizing variants of goodhart's law. *arXiv preprint arXiv:1803.04585*.

Sewon Min, Keshav Krishna, Xinyuan Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. 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*, pages 12076–12100.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10862–10878, Bangkok, Thailand. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. 2022. The effects of reward misspecification: Mapping and mitigating misaligned models.

Arjun Panickssery, Samuel R Bowman, and Shi Feng. 2024. Llm evaluators recognize and favor their own generations. *arXiv preprint arXiv:2404.13076*.

Rafael Rafailov, Arjun Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36.

Varun Raina, Siddharth Krishnamurthy, and Mark Gales. 2024. Llm assisted labeling: An exploration of quality versus cost tradeoffs. *arXiv preprint arXiv:2401.15265*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3982–3992.

Paul Roit, Johan Ferret, Lior Shani, Roee Aharoni, Geoffrey Cideron, Robert Dadashi, Matthieu Geist, Sertan Girgin, Léonard Hussenot, Orgad Keller, Nikola Momchev, Sabela Ramos, Piotr Stanczyk, Nino Vieillard, Olivier Bachem, Gal Elidan, Avinatan Hassidim, Olivier Pietquin, and Idan Szpektor. 2023. Factually consistent summarization via reinforcement learning with textual entailment feedback. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6252–6272. Association for Computational Linguistics.

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

Yuxuan Shen and 1 others. 2026. Rethinking rubric generation for improving llm judge and reward modeling for open-ended tasks. *arXiv preprint arXiv:2602.05125*.

Prasann Singhal, Tanya Forber, Mega Kuo, and Jerry Kreiman. 2023. A long way to go: Investigating length correlations in rlhf. *arXiv preprint arXiv:2310.03716*.

Joar Skalse, Nils Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward hacking. In *Advances in Neural Information Processing Systems*, volume 35.

Peiyi Wang, Lei Li, Liang Chen, Feifan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Yan Song, and Xuandong Zhao. 2024. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.

An Yang, Bowen Yu, Beidi Zhou, Baosong Hui, Bo Zheng, Bowen Zhou, and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. In *International Conference on Machine Learning*.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. 2026. Learning to reason without external rewards.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanzhao Wu, Yonghao Zhuang, Zi Lin, Zhouhan Li, Dacheng Li, and Eric Xing. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36.

Yutong Zuo, Yuxuan Li, Xiang Feng, Wenjia Qiu, Xipeng Wang, and Xipeng Qiu. 2025. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*.

A Implementation Details

NLI verifier: `cross-encoder/nli-deberta-v3-large`, chunked at 32 pairs ($\sim$0.3s each). Embeddings: all-MiniLM-L12-v2 (Reimers and Gurevych, 2019), 384-dim, L2-normed. LoRA: rank 16, α=32, dropout 0.05, q/k/v/o targets, 10,092,544 trainable parameters. GRPO: K=5, clip 0.2, KL 0.01, lr 10^{-6} cosine, 100 warmup steps, batch 4, max 512 tokens. Temperatures: $[0.75, 0.85, 0.95, 1.0, 1.0]$. Hardware: 4$\times$ RTX 3090 (24 GB), $\sim$44s/step, $\sim$120 GPU-hours total.

Step	R1	R2	R3	R4	R5
50	51.6	—	—	—	—
100	22.4	—	—	—	—
200	24.0	50.8	49.6	54.0	53.6
400	—	50.4	53.6	47.2	52.0
600	—	46.4	50.0	47.2	54.8
800	—	52.4	50.0	52.4	59.2
1000	—	50.8	46.4	44.4	54.4

Table 4: Full Arena-Hard GPT-4o-mini win rates (%) across all checkpoints. R1 was terminated after step 100 due to collapse. "—" indicates not evaluated.

Step	R2	R3	R4	R5
200	50.0	**57.2**	46.8	51.2
400	48.8	53.6	45.2	—*
600	48.4	48.4	49.2	58.4
800	48.8	47.2	50.8	49.2
1000	49.2	50.0	43.2	50.8

Table 5: Full Arena-Hard GPT-4.1 win rates (%) across all checkpoints. *R5 step 400 evaluation returned all-TIE (API failure); excluded. "—" otherwise indicates not evaluated.

B Full Win Rate Tables

C Checkpoint Robustness Across Benchmarks

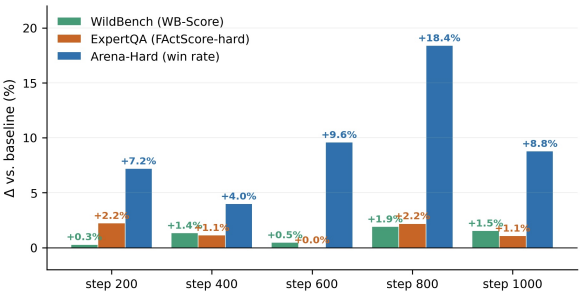


Figure 5: Relative improvement over the base model across ExpertQA, WildBench, and Arena-Hard.

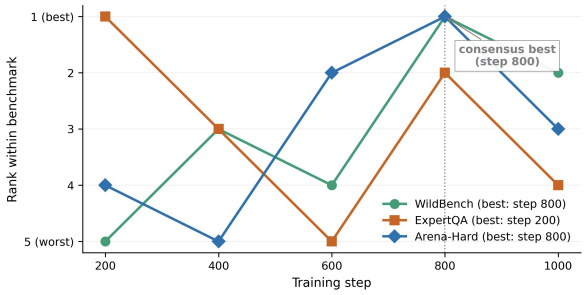


Figure 6: Rank stability across checkpoints. Lower rank is better; rank 1 denotes the best checkpoint within a benchmark.

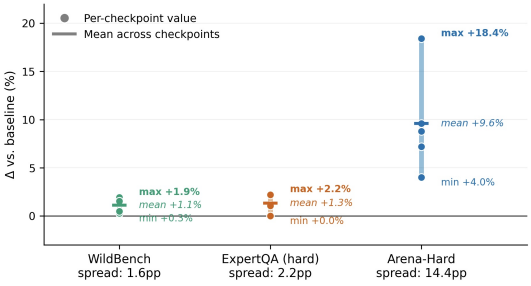


Figure 7: Spread of benchmark performance across checkpoints, measured as improvement over the base model. Larger spread indicates greater sensitivity to checkpoint choice.

D Training Dynamics

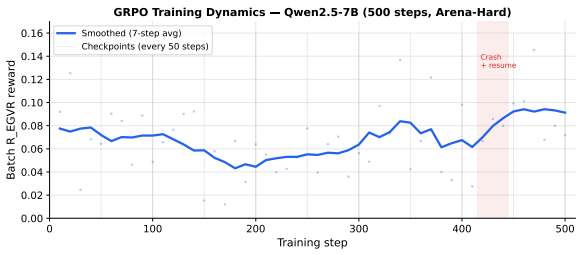


Figure 8: Per-step reward with 7-step smoothing. Run 2 with group-level R_{caus} shows near-zero loss, while Run 3 with per-rollout R_{caus} shows normal GRPO dynamics.

E Reproducibility

Three seeds for Run 1 through step 150 yield step-100 win rates of 22.4%, 23.1%, and 22.9% ($\pm$1.2pp). Post-150 trajectory shape is mildly seed-dependent.

F Bootstrap Confidence Intervals

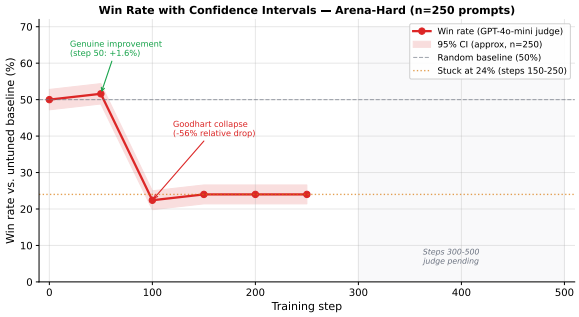


Figure 9: Bootstrap 95% CIs (n=250, 10K resamples). Step-50 interval (45.5, 57.7)% and step-100 interval (17.4, 28.6)% do not overlap.

G Length-Controlled Re-analysis

Truncating step-50 responses to match step-100 average length returns TTR, repetition rate, and self-similarity to their post-collapse values.

H R_{caus} Distribution

Across 250 Arena-Hard prompts, R_{caus} has mean 0.389 and standard deviation 0.138, with 76% of values above 0.30. Entity swap via spaCy NER succeeded on 25/250 prompts; an LLM-based fallback handled the remainder. The spread confirms that R_{caus} discriminates across prompts rather than returning a near-constant signal, validating its use as a training signal that varies across the rollout group.