# REINFORCEMENT LEARNING WITH CONDITIONAL EXPECTATION REWARD

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Reinforcement Learning with Verifiable Rewards (RLVR) has proven effective in enhancing the reasoning capabilities of large language models, particularly in domains such as mathematics where reliable rule-based verifiers can be constructed. However, the reliance on handcrafted, domain-specific verification rules substantially limits the applicability of RLVR to general reasoning domains with free-form answers, where valid answers often exhibit significant variability, making it difficult to establish complete and accurate rules. To address this limitation, we propose Conditional Expectation Reward (CER), which leverages the large language model itself as an implicit verifier, and is therefore applicable to general domains and eliminates the need for external verifiers or auxiliary models. CER is defined as the expected likelihood of generating the reference answer conditioned on the generated answer. In contrast to rule-based verifiers that yield binary feedback, CER provides a soft, graded reward signal that reflects varying degrees of correctness, making it better suited to tasks where answers vary in correctness. Experimental results demonstrate that CER is effective across a wide range of reasoning tasks, spanning both mathematical and general domains, indicating that CER serves as a flexible and general verification mechanism.

## 1 INTRODUCTION

Reinforcement Learning with Verifiable Rewards (RLVR) has demonstrated strong effectiveness in incentivizing the reasoning capabilities of large language models (LLMs), which relies on a verifier to provide accurate and checkable reward signals during learning (Zhou et al., 2025). Such a verifier evaluates the correctness of a generated answer with respect to a given reference answer, and is typically implemented using carefully designed, domain-specific rules that enable deterministic verification (Guo et al., 2025; He et al., 2025). RLVR is particularly useful in mathematical reasoning tasks (Guo et al., 2025; Team et al., 2025), where answers admit canonical or easily normalized representations, allowing correctness to be verified reliably through exact matching or symbolic equivalence checks (Hugging Face, 2025).

However, RLVR remains difficult to extend to broader reasoning domains such as physics, chemistry, finance, and other domains with open-form answers (Ma et al., 2025; Zhou et al., 2025; Liu et al., 2025). In these domains, valid answers often exhibit diverse surface forms and substantial semantic variation, making it challenging to specify exhaustive verification rules. Consequently, constructing reliable verifiers becomes costly or even infeasible, which substantially constrains the applicability of RLVR to narrowly scoped tasks with well-defined answer spaces.

Moreover, rule-based verifiers typically provide binary feedback, assigning rewards only to strictly equivalent answers while treating all other answers as equally incorrect. As a result, they are unable to assign positive rewards to answers that are partially correct, thereby providing limited learning signals during optimization.

To address these issues, we propose Conditional Expectation Reward (CER) to extend RLVR to general reasoning domains. Rather than relying on external verification rules or auxiliary verifier models, CER uses the LLM itself as an implicit verifier. By exploiting the model's internal consistency with respect to a reference answer, CER provides a model-intrinsic reward signal that remains applicable even when explicit verification is unavailable.



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.

Specifically, CER measures the expected probability of generating the reference answer conditioned on the model's generated answer, thereby producing a soft, graded reward signal to verify the generated answer. The underlying intuition is that when a generated answer is identical to, or strongly consistent with, the reference answer, the model will assign a higher conditional probability to reproducing the reference answer given that generation.

We further show that CER can be theoretically interpreted as a smooth relaxation of the exact-match criterion, yielding reward values that reflect varying degrees of consistency between the generated and reference answers. This characteristic is particularly well suited to general reasoning domains, where partial correctness, semantic overlap, and multiple valid surface realizations are common.

We finally conduct experiments to demonstrate the effectiveness of CER. We show that CER achieves great performance on general domains, both mathematical and non-mathematical domains. These findings highlight CER as a general and robust reward mechanism for RLVR, offering a practical solution for extending reinforcement learning to a wide range of reasoning domains.

## 2 Conditional Expectation Reward

We first introduce RLVR, followed by the definition and theoretical properties of CER. We then present the empirical formulation of CER and the corresponding training objective, and finally describe an efficient procedure for computing CER.

**RLVR** RLVR is a reinforcement learning paradigm in which the reward signal is objectively and automatically checkable by a verifier (Zhang et al., 2025). Specifically, for a given question $q$ with a unique reference answer $a^* = a^*(q)$, the policy model $\pi_\theta$ autoregressively generates a solution $s$ and a final answer $a$ to address the question. Here, the solution $s$ does not include the final answer $a$. Owing to the autoregressive factorization, the policy model $\pi_\theta$ satisfies

$$\pi_\theta(a, s|q) = \pi_\theta(s|q)\pi_\theta(a|s, q). \tag{1}$$

This process yields a quadruple $(q, s, a, a^*)$. An illustrative example is provided in Box 1.

RLVR is formulated as the optimization of the following expected reward:

$$\mathcal{L}_f(\theta) = \mathbb{E}_{q\sim\mathcal{D},(s,a)\sim\pi_\theta(\cdot|q)}[f(a, a^*(q))], \tag{2}$$

where $\mathcal{D}$ is the distribution of $q$, and $f(a, a^*(q))$ is a reward that evaluates the correctness of the generated answer $a$ with respect to the reference answer $a^*$ associated with question $q$, which is computed by a verifier $f(\cdot, \cdot)$. In practice, such a verifier is often implemented as a set of carefully designed rules (Hugging Face, 2025). Rule-based verifiers are particularly effective in domains such as mathematics and code generation, where answers admit unambiguous representations and equivalence can be precisely defined.

However, extending rule-based verification to general reasoning domains remains challenging (Ma et al., 2025; Zhou et al., 2025). In these domains, valid answers are often free-form and exhibit substantial variation. Consequently, it is difficult to design a rule-based verifier that is both complete and accurate, which limits the direct applicability of RLVR beyond domains with well-structured and formally verifiable answer spaces.

We illustrate this limitation with a concrete example in Box 2.. Consider the following question, for which multiple answers may be semantically equivalent despite differing in surface form. A typical rule-based verifier would only treat $a_1$ as correct, while assigning zero reward to other valid answers such as $a_2$ and $a_3$. This behavior collapses semantically correct but lexically different answers into the same category as incorrect ones, leading to overly sparse and noisy reward signals. Such rigid verification discourages exploration of diverse yet correct answers and hampers effective learning in general reasoning settings.

**Definition** To generalize RLVR to general domains, we propose the CER, which leverages the policy model itself as an implicit verifier, without relying on external verifiers or models. Instead of explicitly checking answer correctness, CER evaluates the internal consistency of the policy model with respect to a reference answer, thereby enabling applicability to general domains.

For a quadruple $(q, s, a, a^*)$, we define CER as:

$$\begin{aligned}\rho(a, a^*) :=& \mathbb{E}_{s' \sim \pi_\theta(\cdot|q)} \left[ \pi_\theta(a^*|s', q) \mid A = a \right] \\ =& \mathbb{E}_{s' \sim \pi_\theta(\cdot|q,a)} \left[ \pi_\theta(a^*|s', q) \right]. \end{aligned} \tag{3}$$

CER measures the expected likelihood of generating the reference answer $a^*$ given the condition that the model has generated an answer $a$. The intuition is that if the generated answer $a$ is identical to, or strongly correlated with, the reference answer $a^*$, then the policy model should assign a higher conditional probability to generating $a^*$ after having produced $a$.

By optimizing the policy model with respect to CER, the model is encouraged to place higher probability mass on answers that are internally consistent with the reference answer, thereby implicitly guiding the generation process toward $a^*$ without requiring explicit verification.

We illustrate CER with an example from training in Box 3. For the given question, the model generates three distinct answers, $\{14, 13, 94\}$. The answer $14$ receives the highest CER value (0.752), as it exactly matches the reference answer. The answer $13$ attains the second-highest CER value (0.313), reflecting its numerical proximity to the reference answer $14$. In contrast, the answer $94$ receives a near-zero CER value (0.00004), as it is numerically distant from the reference answer.

**Properties** We summarize several fundamental properties of CER, which demonstrate the effectiveness of CER.

- **Boundedness.**

  $$0 \leq \rho(a, a^*) \leq 1.$$

  Since $\pi_\theta(a^*|s', q) \in [0, 1]$ for all $(q, s')$, the weighted sum of these probabilities also lies in $[0, 1]$. As a result, CER provides a bounded and well-scaled reward signal that is suitable for stable optimization.

- **Minimum.**

  $$\rho(a, a^*) = 0 \iff \pi_\theta(a^*|s', q) = 0 \quad \text{for all } (q, s') \text{ with } \pi_\theta(s'|q, a) > 0.$$

  $\rho(a, a^*)$ is zero exactly when the policy assigns probability 0 to $a^*$ on any $(q, s')$ that appear in the conditional distribution defined by $A = a$. In this case, once the policy has generated $a$, it is impossible for the model to regenerate $a^*$ under the same posterior distribution over $(q, s')$.

- **Maximum.**

  $$\rho(a, a^*) = 1 \iff \pi_\theta(a^*|s', q) = 1 \quad \text{for all } (q, s') \text{ with } \pi_\theta(s'|q, a) > 0.$$

  $\rho(a, a^*)$ reaches its maximum value only when the policy assigns probability 1 to the reference answer $a^*$ for every $(q, s')$ that can occur under the conditional distribution defined by $A = a$. In this case, the policy cannot produce any alternative answer with positive probability. Consequently,

  $$\rho(a, a^*) = 1 \implies a = a^*$$

- **Self-Consistency.**

  **Theorem 1** (Exact-Match Case)**.** *If $a = a^*$, then*

  $$\begin{aligned}\rho(a^*, a^*) =& \mathbb{E}_{s' \sim \pi_\theta(\cdot|q)} \left[ \pi_\theta(a^*|s', q) \mid A = a^* \right] \\ =& \mathbb{E}_{s' \sim \pi_\theta(\cdot|q,a^*)} \left[ \pi_\theta(a^*|s', q) \right] \\ \geq& \mathbb{E}_{s \sim \pi_\theta(\cdot|q)} \left[ \pi_\theta(a^*|s, q) \right]. \end{aligned}$$

  *with equality if and only if $\pi_\theta(a^*|s, q)$ is constant over all $(q, s)$ such that $\pi_\theta(s|q) > 0$.*

  See Appendix A.2 for the proof. This shows that conditioning on the event that the policy has generated $a^*$ strictly increases the posterior predictive probability of regenerating $a^*$, unless the policy assigns an identical likelihood to $a^*$ across all $(q, s)$ pairs with $\pi_\theta(s|q) > 0$. Therefore, CER exhibits a self-consistency amplification effect via posterior reweighting toward higher probability assigned to the reference answer in the exact-match case.

- **Equivalence.**

  **Theorem 2** (Value Equivalence)**.**

$$\begin{aligned}\mathcal{L}_\rho(\theta) &= \mathbb{E}_{q\sim\mathcal{D},(s,a)\sim\pi_\theta(\cdot|q)}[\rho(a, a^*(q))] \\ &= \mathbb{E}_{q\sim\mathcal{D},(s,a)\sim\pi_\theta(\cdot|q)}[\mathbb{I}(a = a^*(q))],\end{aligned}$$

  *i.e., the expected CER objective is equivalent in value to the exact-match objective, where* $\mathbb{I}(a = a^*(q))$ *indicates whether* $a$ *exactly matches* $a^*(q)$*.*

  Thus, $\rho(a, a^*)$ can be interpreted as a soft generalization of the hard exact-match reward $\mathbb{I}(a = a^*)$, while preserving the same expected value. CER yields a continuous-valued reward rather than a binary signal. This property allows CER to provide graded feedback that reflects varying degrees of consistency between the generated answer $a$ and the reference answer $a^*$, which is particularly beneficial in general domains where partial correctness or semantic similarity may be present.

In summary, these properties show that CER is a well-behaved and principled reward function. It is bounded and properly scaled, admits clear minimum and maximum conditions, and exhibits a self-consistency amplification effect when the generated answer matches the reference answer. Moreover, CER is value-equivalent in expectation to the exact-match objective while providing a continuous, graded reward signal, thereby serving as a soft generalization of exact-match rewards in general domains.

**Empirical CER** We next derive an empirical form of CER, as the definition in Eq. (3) is intractable due to the summation over all possible outcomes under $\pi_\theta(a^*|s', q)$. To obtain a computable approximation, we apply Bayes' rule and Monte Carlo sampling to derive an empirical estimator of CER:

$$\begin{aligned}\rho(a, a^*) =& \mathbb{E}_{s'\sim\pi_\theta(\cdot|q)}\left[\pi_\theta(a^*|s', q) \,\middle|\, A = a\right] \\ =& \sum_{s'} \pi_\theta(s'|q, a)\, \pi_\theta(a^*|s', q) \\ =& \frac{\sum_{s'} \pi_\theta(s'|q)\pi_\theta(a|s', q)\pi_\theta(a^*|s', q)}{\sum_{s''} \pi_\theta(s''|q)\pi_\theta(a|s'', q)} \\ \approx& \frac{\sum_{j=1}^{M} \pi_\theta(a|s_j, q)\, \pi_\theta(a^*|s_j, q)}{\sum_{j=1}^{M} \pi_\theta(a|s_j, q)} \\ & \text{where } s_j \sim \pi_\theta(\cdot|q). \end{aligned} \tag{4}$$

The third line applies Bayes' rule to rewrite $\pi_\theta(s'|q, a)$ in terms of quantities compatible with the autoregressive factorization of LLMs.

The final line further applies Monte Carlo estimation by drawing $M$ independent samples $s_j \sim \pi_\theta(\cdot|q)$ to produce an empirical estimate of CER.

The resulting empirical CER is a normalized likelihood-weighted average, where each term $\pi_\theta(a^*|s_j, q)$ is weighted by $\pi_\theta(a|s_j, q)$. This weighting captures the joint consistency of $a$ and $a^*$ under the same conditional context, so that samples for which the policy assigns high probability to both $a$ and $a^*$ contribute more to the estimator, leading to a larger value of CER.

**Objective** We finally define the objective based on the empirical CER. For a quadruple $(q, s, a, a^*)$, the reward function is defined as:

$$\begin{aligned} R(q, s, a, a^*) &= \frac{\sum_{j=1}^{M} \pi_\theta(a|s_j, q)\pi_\theta(a^*|s_j, q)}{\sum_{j=1}^{M} \pi_\theta(a|s_j, q)}, \\ &\text{where } s_j \sim \pi_\theta(\cdot|q). \end{aligned} \tag{5}$$

Using this reward function, we define the training objective as

$$\begin{aligned}\mathcal{L}_\rho(\theta) &= \mathbb{E}_{q\sim\mathcal{D},(s,a)\sim\pi_\theta(\cdot|q)}[\rho(a, a^*)] \\ &\approx \mathbb{E}_{q\sim\mathcal{D},(s,a)\sim\pi_\theta(\cdot|q)}[R(q, s, a, a^*)]. \end{aligned} \tag{6}$$

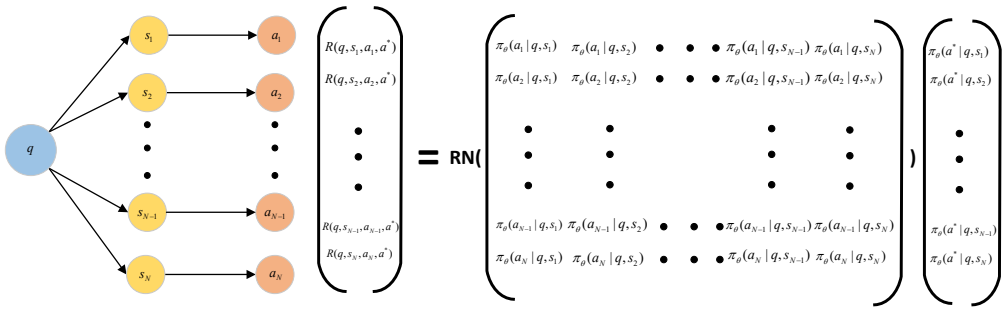


Figure 1: An illustration of CER computation, where RN(·) denotes row normalization. The left panel depicts the generation process of the quadruple $(q, s_i, a_i, a^*)$, while the right panel shows the CER computation for the quadruple, corresponding to Eq. (8).

Then the corresponding policy gradient is given by

$$\begin{aligned}\nabla_\theta \mathcal{L}_\rho(\theta) \approx & \mathbb{E}_{q\sim\mathcal{D},(s,a)\sim\pi_\theta(\cdot|q)}[R(q,s,a,a^*)\nabla_\theta \log \pi_\theta(a,s|q)] \\ \approx & \mathbb{E}_{q\sim\mathcal{D}}[\frac{1}{N}\sum_{i=1}^{N} R(q,s_i,a_i,a^*)\nabla_\theta \log \pi_\theta(a_i,s_i|q)], \\ & \text{where } (s_i,a_i)\sim\pi_\theta(\cdot|q). \end{aligned} \tag{7}$$

For each $q$, we sample $N$ independent $(s_i, a_i)$ from $\pi_\theta(\cdot|q)$ for estimating the gradient. The reward $R(q, s_i, a_i, a^*)$ is treated as a fixed scalar with respect to $\theta$ during optimization to detach it from gradient computation for stable policy learning (Ziegler et al., 2019; Ouyang et al., 2022).

**Efficiency** We now describe an efficient procedure for computing CER by reusing samples, avoiding redundant computations and adjusting the hyperparameter. As shown in Eq. (5), computing CER requires sampling $M$ independent solutions $\{s_j\}_{j=1}^M$ from $\pi_\theta(\cdot|q)$. However, CER can be seamlessly integrated into policy gradient without additional sampling. Specifically, for each question $q$, we already sample $N$ independent $\{s_i\}_{i=1}^N$ from $\pi_\theta(\cdot|q)$ to estimate the policy gradient. These same samples can be directly reused for reward computation by setting $\{s_j\}_{j=1}^M := \{s_i\}_{i=1}^N$. As a result, CER incurs no extra sampling cost.

To understand the computation of CER more intuitively, we further show that the CER computing can be expressed in a tensorized form. Let $M := N$ and $\{s_j\}_{j=1}^M := \{s_i\}_{i=1}^N$, and define $\boldsymbol{R} \in [0,1]^N$ with entries $\boldsymbol{R}_i = R(q, s_i, a_i, a^*)$, $\boldsymbol{W} \in [0,1]^{N\times M}$ with entries $\boldsymbol{W}_{ij} = \pi_\theta(a_i|s_j, q)$, and $\boldsymbol{P} \in [0,1]^M$ with entries $\boldsymbol{P}_j = \pi_\theta(a^*|s_j, q)$. The reward vector $\boldsymbol{R}$ can then be written as

$$\boldsymbol{R} = \boldsymbol{D}^{-1}\boldsymbol{W}\boldsymbol{P}, \tag{8}$$

where $\boldsymbol{D}$ is a diagonal matrix whose entries are the row sums of $\boldsymbol{W}$. See Figure 1 for an illustration.

Although this approach avoids additional sampling, directly computing Eq. (8) still requires $M(N+1)$ forward passes to compute the entries of $\boldsymbol{W}$ and $\boldsymbol{P}$. We further reduce this cost by eliminating redundant computations. In particular, if two sampled answers $a_{i_1}$ and $a_{i_2}$ are identical, then their corresponding rewards satisfy $R(q, s_{i_1}, a_{i_1}, a^*) = R(q, s_{i_2}, a_{i_2}, a^*)$, and thus the reward only needs to be computed once for each unique answer.

Besides, the hyperparameter $M$ controls a trade-off between computational efficiency and reward estimation accuracy. Larger values of $M$ yield more accurate estimates of CER at the cost of increased computation, while smaller values improve efficiency with a loss in precision. We can adjust $M$ to achieve a balance between performance and efficiency.

## 3 Related Work

**RLVR** RLVR (Lambert et al., 2024; Guo et al., 2025; Team et al., 2025) has emerged as an effective paradigm for improving the reasoning performance of LLMs. It typically relies on rule-based verifiers to provide accurate and stable reward signals, such as the math-verify library (Hugging Face, 2025) for mathematical reasoning (Guo et al., 2025) and the SandboxFusion toolbox (Cheng et al., 2024) for code generation (Luo et al., 2025; He et al., 2025). These methods are particularly effective in domains with unambiguous answers and deterministic equivalence rules. However, their applicability is limited in general reasoning domains, where valid answers are often open-form and exhibit substantial surface variation. CER aims to extend RLVR to such general domains.

**General Domains** Existing verification methods for general domains can be broadly categorized into model-based and perplexity-based verifiers.

Model-based verifiers use a fine-tuned LLM to evaluate the correctness of a generated answer with respect to a reference answer. For example, Kimi-k1.5 (Team et al., 2025) fine-tunes a model on large-scale verification data to acquire verification capabilities. General-Verifier (Ma et al., 2025) further develops a generative verifier trained specifically for chain-of-thought answer verification, enabling more context-aware judgments.

Perplexity-based verifiers instead define rewards using the likelihood or perplexity of the reference answer under the language model. VeriFree (Zhou et al., 2025) combines perplexity-based rewards with variance reduction techniques to construct a training objective. Building on this line of work, Nover (Liu et al., 2025) introduces length normalization to mitigate length bias, while RLPR (Yu et al., 2025) reformulates perplexity as a sum of token-level probabilities to further reduce sensitivity to answer length.

In contrast to both approaches, CER leverages self-consistency between generated answers and the reference answer to produce soft, graded, and model-intrinsic reward signals. This design enables reliable feedback without requiring additional verifier models or handcrafted rules, making CER broadly applicable to general reasoning domains.

## 4 Experiments

We first describe the experimental settings in Section 4.1. We then present the main experimental results in Section 4.2 to evaluate the effectiveness of the proposed method. Next, we analyze the computational efficiency of CER in Section 4.3. Finally, in Section 4.4, we provide a detailed visualization of the CER computation process to offer further insights into its behavior and properties. Please refer to Appendix A.3 for more experimental results.

### 4.1 Settings

**Datasets** We evaluate the performance of CER across both mathematical and general reasoning domains. Accordingly, we train models on two datasets: the mathematical dataset MATH-7.5K (Hendrycks et al., 2021) and the general-domain dataset WebInstruct (Ma et al., 2025). For WebInstruct, we retain only non-mathematical questions at the university difficulty level to focus on general-domain beyond mathematics, yielding a dataset of 50K questions. WebInstruct includes domains such as physics, chemistry, biology, finance and so on. This subset spans a wide range of disciplines, such as physics, chemistry and biology.

**Evaluation** We evaluate the models on four mathematical datasets, MATH500 (Lightman et al., 2023), AMC23 (Art of Problem Solving, 2025b), AIME2024 and AIME2025 (Art of Problem Solving, 2025a), and two general-domain datasets, SuperGPQA (Du et al., 2025) and MMLU-Pro (Wang et al., 2024). Performance is reported using the pass@1 metric. For the mathematical datasets, pass@1 is computed using a rule-based verifier (Hugging Face, 2025). For the general-domain datasets, which consist of multiple-choice questions, pass@1 is computed via exact matching. For each mathematical dataset, we conduct 16 evaluation runs and report the average performance.

Table 1: The performance of models trained on a general-domain dataset.

| Methods | MATH500 | AMC23 | AIME2024 | AIME2025 | MMLU-Pro | SuperGPQA | Average |
|---|---|---|---|---|---|---|---|
| *Qwen3-4B-Base* | | | | | | | |
| Base | 62.6 | 40.5 | 10.6 | 8.1 | 42.0 | 21.0 | 30.8 |
| Exact-Match | 78.0 | 61.4 | 22.3 | 20.6 | 62.9 | 33.5 | 46.5 |
| Rule-based | 84.5 | 65.8 | 21.7 | 18.5 | 62.3 | 32.6 | 47.6 |
| VeriFree | 83.4 | 62.2 | 19.6 | 16.7 | 58.7 | 29.1 | 44.9 |
| RLPR | 80.4 | 64.1 | 19.1 | 17.5 | 62.5 | 33.5 | 46.2 |
| Nover | 81.3 | 65.7 | 21.4 | 19.8 | 62.7 | 34.1 | 47.5 |
| General-verifier | 84.6 | 64.1 | 21.9 | 17.7 | 63.7 | 34.2 | 47.7 |
| CER | 82.6 | **66.7** | **22.8** | **21.7** | 64.7 | **36.0** | 49.0 |
| CER+Rule | **85.6** | 66.5 | 21.5 | 21.5 | **64.9** | 35.8 | **49.3** |
| *Qwen3-8B-Base* | | | | | | | |
| Base | 73.9 | 53.1 | 14.6 | 12.3 | 51.9 | 27.0 | 38.9 |
| Exact-Match | 82.4 | 66.6 | 25.4 | 20.4 | 66.2 | 36.2 | 49.5 |
| Rule-based | 86.0 | 72.0 | **26.7** | 21.0 | 66.6 | 37.7 | 51.7 |
| VeriFree | 86.0 | 61.4 | 22.3 | 19.6 | 65.4 | 35.5 | 48.4 |
| RLPR | 87.0 | 71.6 | 22.3 | 19.6 | 66.8 | 35.1 | 50.4 |
| Nover | 84.2 | 71.6 | 26.7 | 17.5 | 67.9 | 37.0 | 50.8 |
| General-verifier | 84.8 | 74.1 | 25.0 | **21.7** | 67.3 | 37.7 | 51.8 |
| CER | **87.2** | 72.3 | 25.8 | 20.6 | 69.7 | **38.5** | 52.4 |
| CER+Rule | 85.7 | **76.4** | 24.1 | 21.0 | **71.1** | 38.2 | **52.8** |

Table 2: The performance of models trained on a mathematical dataset.

| Methods | MATH500 | AMC23 | AIME2024 | AIME2025 | MMLU-Pro | SuperGPQA | Average |
|---|---|---|---|---|---|---|---|
| *Qwen3-4B-Base* | | | | | | | |
| Base | 62.6 | 40.5 | 10.6 | 8.1 | 42.0 | 21.0 | 30.8 |
| Exact-match | 81.2 | 57.3 | 17.7 | 15.6 | 46.7 | 24.2 | 40.5 |
| Rule-based | 84.2 | 63.1 | 22.9 | **21.5** | **62.5** | **32.2** | 47.7 |
| VeriFree | 81.5 | 62.7 | 19.8 | 18.1 | 59.9 | 30.9 | 45.5 |
| RLPR | 82.4 | 58.1 | 21.7 | 17.9 | 60.7 | 32.8 | 45.6 |
| Nover | 83.0 | 62.8 | 21.9 | 17.5 | 59.5 | 31.8 | 46.1 |
| General-verifier | 83.6 | 63.0 | 19.8 | 19.0 | 58.5 | 30.9 | 45.7 |
| CER | 84.1 | 63.6 | **24.8** | 20.0 | 60.8 | 32.1 | 47.6 |
| CER+Rule | **85.0** | **67.5** | 23.3 | 20.8 | 61.2 | 31.3 | **48.2** |
| *Qwen3-8B-Base* | | | | | | | |
| Base | 73.9 | 53.1 | 14.6 | 12.3 | 51.9 | 27.0 | 38.9 |
| Exact-match | 80.0 | 63.4 | 16.5 | 13.5 | 61.4 | 33.5 | 44.7 |
| Rule-based | 86.7 | 70.2 | 26.3 | 22.7 | **65.8** | 35.1 | 51.1 |
| VeriFree | 85.0 | 68.4 | 22.9 | 20.2 | 62.3 | 32.4 | 48.5 |
| RLPR | 85.8 | 67.2 | 21.7 | 20.9 | 65.4 | 34.3 | 49.2 |
| Nover | 87.2 | 69.9 | 22.2 | 22.2 | 63.9 | 34.1 | 49.9 |
| General-verifier | 86.0 | 69.4 | **26.7** | 20.6 | 64.3 | 34.9 | 50.3 |
| CER | 87.2 | 70.9 | 23.8 | **23.1** | 64.8 | 35.0 | 50.8 |
| CER+Rule | **87.3** | **72.0** | 26.5 | 21.0 | 65.6 | **36.0** | **51.4** |

**Baselines** We compare CER with several baseline verifiers. These include the exact-match verifier, which checks whether the generated answer exactly matches the reference answer; a model-based verifier, General-verifier (Ma et al., 2025), which employs an external LLM to assess answer correctness; and three perplexity-based verifier, VeriFree (Zhou et al., 2025), RLPR (Yu et al., 2025) and Nover (Liu et al., 2025), which uses the perplexity of the reference answer for verification, a rule-based verifier (Hugging Face, 2025), which verifies the correctness by utilizing handcrafted rules.

### 4.2 Results

**CER demonstrates strong generality across domains.** On the general-domain training dataset (Table 1), CER achieves the best average performance among all compared methods for both Qwen3-4B-Base and Qwen3-8B-Base, except for the combined method CER+Rule. In particular, CER consistently outperforms exact-match rewards, the perplexity-based reward VeriFree, and learned verifiers such as General-verifier across most benchmarks. Its advantage is especially pronounced on general-domain evaluations such as MMLU-Pro and SuperGPQA. Notably, these gains are achieved without relying on domain-specific handcrafted rules or external verifier models.

When trained on the mathematical dataset (Table 2), CER achieves performance comparable to rule-based rewards while outperforming learned verifier approaches. Despite the absence of an external verifier, CER maintains strong performance across mathematical benchmarks, indicating that it does not overfit to a specific domain.

Table 3: The average performance across six datasets and corresponding runtime.

| Model | Performance | Runtime |
|---|---|---|
| Exact-match | 46.5 | 45.2h |
| Rule | 47.6 | 54.7h |
| VeriFree | 44.9 | 58.7h |
| RLPR | 46.2 | 59.5h |
| Nover | 47.5 | 60.4h |
| General-verifier | 47.7 | 57.5h |
| CER (M=1) | 46.6 | 47.8h |
| CER (M=2) | 48.0 | 53.1h |
| CER (M=4) | 48.2 | 56.4h |
| CER (M=8) | 48.6 | 61.1h |
| CER (M=16) | 49.0 | 67.4h |

Overall, these results suggest that CER can serve as a unified reward formulation for both general-domain and mathematics-oriented reasoning tasks.

**CER substantially improves over exact-match rewards.** As established in Theorem 2, CER can be viewed as a soft generalization of exact-match rewards. While exact-match provides only binary supervision, CER assigns continuous-valued rewards that reflect partial correctness and semantic similarity to the reference answer. Empirically, this leads to consistent improvements across datasets and model scales. The graded feedback provided by CER yields denser and more informative learning signals, especially when correct answers admit surface-level variations.

**CER is complementary to rule-based rewards.** We further investigate a simple combination strategy in which the final reward is defined as the average of the CER score and the rule-based reward. As shown in Tables 1 and 2, the combined method (CER+Rule) generally outperforms either approach used alone, demonstrating their complementary strengths.

On the general-domain dataset, CER enhances rule-based rewards by providing graded supervision. Rule-based methods assign positive reward only to strictly equivalent answers and fail to distinguish partially correct outputs, resulting in sparse learning signals. CER alleviates this limitation by assigning non-binary rewards that better reflect answer quality.

On the mathematical dataset, rule-based rewards complement CER by more reliably capturing mathematical equivalence, thereby correcting errors introduced by imperfect similarity estimation. Overall, these results highlight the benefits of combining CER with rule-based rewards across different domains.

### 4.3 Efficiency

We analyze the computational efficiency of various rewards and their corresponding performance. In CER, the computational cost is governed by the hyperparameter $M$ in Eq. (5), which specifies the number of samples used to estimate the reward. **By tuning $M$, CER enables a flexible mechanism to balance runtime efficiency and reward fidelity.**

Table 3 reports runtime and average performance across methods, with all experiments conducted on four NVIDIA H100 GPUs. For CER, increasing $M$ improves performance at the cost of higher runtime overhead. Empirically, CER exhibits a smooth and controllable trade-off, enabling practitioners to select $M$ that balances efficiency and performance under given computational constraints.

Exact-match rewards incur the lowest overhead but yield inferior performance. CER with smaller $M$ and rule-based rewards achieve reasonable performance while remaining efficient, whereas CER with large $M$, VeriFree, and General-verifier incur higher runtime costs due to multiple LLM queries during reward computation.

### 4.4 Visualization of CER Computing

To better illustrate the computation of CER, we visualize the components in Eq. (8). Recall that $\boldsymbol{R}_i$ denotes the CER associated with the quadruple $(q, s_i, a_i, a^*)$, $\boldsymbol{W}_{ij} = \pi_\theta(a_i \mid s_j, q)$ represents

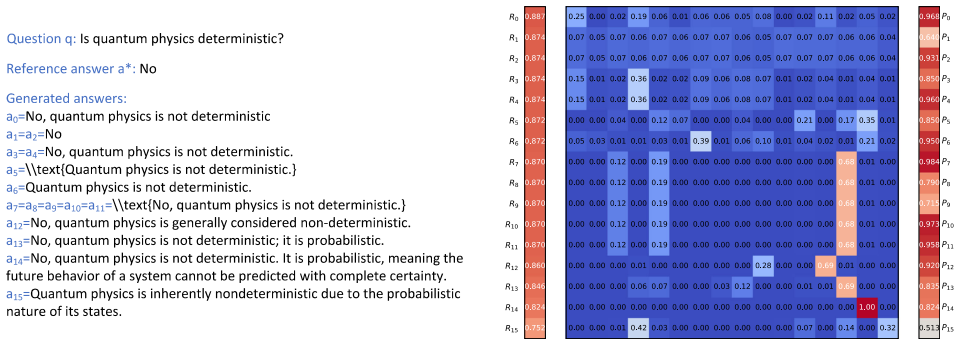


Figure 2: This figure illustrates the computation of CER as defined in Eq. (8). The left panel shows the question, the reference answer, and the 16 generated answers. The right panel depicts the components: the reward vector $\boldsymbol{R}$ (left column), the row-normalized matrix $\boldsymbol{D}^{-1}\boldsymbol{W}$ (central block), and the reference-likelihood vector $\boldsymbol{P}$ (right column).

the likelihood of generating answer $a_i$ conditioned on solution $s_j$ and question $q$, and $\boldsymbol{D}$ is the diagonal matrix whose entries are the row sums of $\boldsymbol{W}$. In addition, $\boldsymbol{P}_j = \pi_\theta(a^* \mid s_j, q)$ measures the likelihood of producing the reference answer $a^*$ given $(s_j, q)$.

Figure 2 presents a training example. The left panel shows the question $q$, the reference answer $a^*$, and the 16 generated answers $\{a_i\}$. The right panel visualizes the vectors $\boldsymbol{R}$ and $\boldsymbol{P}$ together with $\boldsymbol{D}^{-1}\boldsymbol{W}$. Specifically, the left column corresponds to the reward vector $\boldsymbol{R}$, the middle block depicts the normalized matrix $\boldsymbol{D}^{-1}\boldsymbol{W}$, and the right column corresponds to the vector $\boldsymbol{P}$.

Several observations can be drawn from this visualization. **First, CER effectively captures both surface-level variation and semantic similarity among answers, which is especially important for free-form responses.** In this example, the 16 generated answers contain 10 unique surface forms with similar semantics, such as "No, quantum physics is generally considered non-deterministic." and "No, quantum physics is not deterministic." CER assigns positive rewards to all semantically consistent answers, whereas exact-match or rule-based methods would reward only the strictly matching answer "No". This demonstrates that CER provides richer reward signals for general-domain reasoning tasks.

**Second, answers with higher CER rewards tend to align more strongly with solutions that also assign high likelihood to the reference answer**. Since most entries of $\boldsymbol{P}$ are relatively large, this alignment can be observed from the sparsity patterns of $\boldsymbol{D}^{-1}\boldsymbol{W}$. From top to bottom, the rows of the normalized matrix become increasingly sparse, leading to smaller CER values. This trend is consistent with Eq. (5), where reduced overlap with other solutions yields lower rewards.

**Third, the visualization suggests that increasing the value of $M$ in Eq. (5) may improve performance.** A larger $M$ can produce a denser and more stable normalized matrix, leading to more accurate CER estimation.

**Finally, identical answers receive identical CER rewards.** For example, since $a_1 = a_2$, the corresponding rows in the normalized matrix are identical, resulting in equal CER values for $a_1$ and $a_2$. This reflects the consistency of CER with respect to repeated answer instances.

## 5 Conclusion

In this paper, we propose CER as a general framework for extending RLVR beyond domains that rely on strict rule-based verification. By leveraging the LLM itself as an implicit verifier, CER produces soft, graded reward signals that capture partial correctness and semantic consistency, overcoming the limitations of binary feedback. Our theoretical analysis shows that CER can be viewed as a smooth relaxation of exact-match metric, providing a principled connection to conventional verifiable rewards. Empirically, we demonstrate that CER is effective across both mathematical and general-domain reasoning tasks. These results suggest that CER provides a flexible and broadly applicable mechanism for reinforcement learning in LLMs, enabling more general and robust reasoning capabilities.

### AI USE STATEMENT

In this work, we used generative AI tools solely to assist with language editing and proofreading, including improving grammar, clarity, readability, and overall presentation of the manuscript. We did not use generative AI tools to generate research ideas, formulate the methodology, conduct experiments, analyze experimental results, or produce scientific claims, figures, or other research artifacts. All AI-assisted text was carefully reviewed and revised by the authors to ensure that it accurately reflects the underlying research and does not introduce unsupported claims or errors. The authors take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

We have reviewed the ICLR Code of Ethics and identify no specific ethical concerns. Our study does not involve human subjects, personally identifiable or sensitive information, or private data. We use publicly available datasets and open-source models and follow standard practices to ensure research integrity and reproducibility.

### REPRODUCIBILITY STATEMENT

We have taken several steps to facilitate the reproducibility of our results. The proposed CER method and its training procedure are described in detail in the main paper, including the key definitions and algorithmic components. Complete hyperparameter settings and additional experimental details are provided in Appendix A.3. We also provide the source code in the supplementary material, enabling researchers to reproduce our implementation and experiments. Together, these materials provide the necessary methodological, experimental, and implementation details for reproducing our results.

## REFERENCES

Art of Problem Solving. Aime problems and solutions, 2025a. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

Art of Problem Solving. Amc problems and solutions, 2025b. URL https://artofproblemsolving.com/wiki/index.php?title=AMC_Problems_and_Solutions.

Yao Cheng, Jianfeng Chen, Jie Chen, Li Chen, Liyu Chen, Wentao Chen, Zhengyu Chen, Shijie Geng, Aoyan Li, Bo Li, et al. Fullstack bench: Evaluating llms as full stack coders. *arXiv preprint arXiv:2412.00535*, 2024.

Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, et al. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*, 2025.

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

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, et al. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Hugging Face. Math-verify: A robust mathematical expression evaluation system. GitHub repository and Python package, 2025. https://github.com/huggingface/Math-Verify, version 0.8.0.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE samples, get a baseline for free!, 2019.

Ariel Kwiatkowski, Natasha Butt, Ismail Labiad, Julia Kempe, and Yann Ollivier. Likelihood-based reward designs for general llm reasoning. *arXiv preprint arXiv:2602.03979*, 2026.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Wei Liu, Siya Qi, Xinyu Wang, Chen Qian, Yali Du, and Yulan He. Nover: Incentive training for language models via verifier-free reinforcement learning. *arXiv preprint arXiv:2505.16022*, 2025.

Michael Luo, Sijun Tan, Roy Huang, Ameen Patel, Alpay Ariyak, Qingyang Wu, Xiaoxiang Shi, Rachel Xin, Colin Cai, Maurice Weber, et al. Deepcoder: A fully open-source 14b coder at o3-mini level. *Notion Blog*, 2025.

Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhu Chen. General-reasoner: Advancing llm reasoning across all domains. *arXiv preprint arXiv:2505.14652*, 2025.

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

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

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290, 2024.

Yuning Wu, Jiahao Mei, Ming Yan, Chenliang Li, Shaopeng Lai, Yuran Ren, Zijia Wang, Ji Zhang, Mengyue Wu, Qin Jin, et al. Writingbench: A comprehensive benchmark for generative writing. *Advances in Neural Information Processing Systems*, 38, 2026.

Tianyu Yu, Bo Ji, Shouli Wang, Shu Yao, Zefan Wang, Ganqu Cui, Lifan Yuan, Ning Ding, Yuan Yao, Zhiyuan Liu, et al. Rlpr: Extrapolating rlvr to general domains without verifiers. *arXiv preprint arXiv:2506.18254*, 2025.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, et al. A survey of reinforcement learning for large reasoning models. *arXiv preprint arXiv:2509.08827*, 2025.

Xiangxin Zhou, Zichen Liu, Anya Sims, Haonan Wang, Tianyu Pang, Chongxuan Li, Liang Wang, Min Lin, and Chao Du. Reinforcing general reasoning without verifiers. *arXiv preprint arXiv:2505.21493*, 2025.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

# A APPENDIX

## A.1 EXAMPLES

**Box 1:**

**Question $q$:** What is the value of $x$ in the equation $2x + 3 = 7$?
**Solution $s$:** Solve the equation step by step.
First, subtract 3 from both sides:

$$2x + 3 = 7 \;\Rightarrow\; 2x = 4.$$

Next, divide both sides by 2:

$$x = 2.$$

Therefore, the value of $x$ is
**Answer $a$:** 2
**Reference answer $a^*$:** 2

**Box 2:**

**Question $q$:** Is quantum physics deterministic?
**Reference answer $a^*$:** No
**Answer $a_1$:** No
**Answer $a_2$:** Quantum physics is not deterministic.
**Answer $a_3$:** No, quantum physics is not deterministic; it is probabilistic.

**Box 3:**

**Question $q$:** How many positive multiples of 7 that are less than 1000 end with the digit 3?
**Reference answer $a^*$:** 14
**Generated Answers and CER $(a, \rho(a, a^*))$:**
$\{(14, 0.752), (13, 0.313), (94, 0.00004)\}$

## A.2 PROOF

**Theorem 1** (Exact-Match Case)**.** *If $a = a^*$, then*

$$\begin{aligned}\rho(a^*, a^*) =& \mathbb{E}_{s' \sim \pi_\theta(\cdot|q)}\left[\pi_\theta(a^*|s', q) \,\middle|\, A = a^*\right] \\ =& \mathbb{E}_{s' \sim \pi_\theta(\cdot|q, a^*)}\left[\pi_\theta(a^*|s', q)\right] \\ \geq& \mathbb{E}_{s \sim \pi_\theta(\cdot|q)}\left[\pi_\theta(a^*|s, q)\right].\end{aligned}$$

*with equality if and only if $\pi_\theta(a^*|s, q)$ is constant over all $(q, s)$ such that $\pi_\theta(s|q) > 0$.*

*Proof.* Since $\rho(a, a^*)$ is defined for a generated answer $a$, we always have the probability $\Pr(A = a) > 0$. In particular, if $a = a^*$, then $\Pr(A = a^*) > 0$, so conditioning on $A = a^*$ is well defined.

By Bayes' rule,

$$\begin{aligned}&\mathbb{E}_{s' \sim \pi_\theta(\cdot|q)}[\pi_\theta(a^*|s', q)|A = a^*] \\ &= \frac{\mathbb{E}_{s \sim \pi_\theta(\cdot|q)}[\pi_\theta(a^*|s, q)\, \mathbb{I}(A = a^*)]}{\Pr(A = a^*)} \\ &= \frac{\mathbb{E}_{s \sim \pi_\theta(\cdot|q)}[\pi_\theta(a^*|s, q)\ \Pr(A = a^*|q, s)]}{\mathbb{E}_{s \sim \pi_\theta(\cdot|q)}[\Pr(A = a^*|q, s)]} \\ &= \frac{\mathbb{E}_{s \sim \pi_\theta(\cdot|q)}\left[\pi_\theta(a^*|s, q)^2\right]}{\mathbb{E}_{s \sim \pi_\theta(\cdot|q)}[\pi_\theta(a^*|s, q)]}.\end{aligned}$$

Hence,

$$\rho(a^*, a^*) = \frac{\mathbb{E}_{s\sim\pi_\theta(\cdot|q)}\big[\pi_\theta(a^*|s,q)^2\big]}{\mathbb{E}_{s\sim\pi_\theta(\cdot|q)}[\pi_\theta(a^*|s,q)]}.$$

Since $\pi_\theta(a^*|s,q) \geq 0$, Jensen's inequality (or equivalently $\mathbb{E}[X^2] \geq \mathbb{E}[X]^2$) implies

$$\rho(a^*, a^*) \;\geq\; \mathbb{E}_{s\sim\pi_\theta(\cdot|q)}\big[\pi_\theta(a^*|s,q)\big],$$

which proves the desired inequality.

Equality holds if and only if $\pi_\theta(a^*|s,q)$ is constant over all $(q,s)$ such that $\pi_\theta(s|q) > 0$.

□

**Theorem 2** (Value Equivalence)**.**

$$\begin{aligned}\mathcal{L}_\rho(\theta) &= \mathbb{E}_{q\sim\mathcal{D},(s,a)\sim\pi_\theta(\cdot|q)}[\rho(a, a^*(q))] \\ &= \mathbb{E}_{q\sim\mathcal{D},(s,a)\sim\pi_\theta(\cdot|q)}[\mathbb{I}(a = a^*(q))],\end{aligned}$$

*i.e., the expected CER objective is equivalent in value to the exact-match objective, where* $\mathbb{I}(a = a^*(q))$ *indicates whether* $a$ *exactly matches* $a^*(q)$*.*

*Proof.* By definition,

$$\begin{aligned}\mathcal{L}_\rho(\theta) &= \mathbb{E}_{q\sim\mathcal{D},(s,a)\sim\pi_\theta(\cdot|q)}[\rho(a, a^*(q))] \\ &= \mathbb{E}_{q\sim\mathcal{D}}\left[\sum_{s,a} \pi_\theta(s,a|q)\, \rho(a, a^*(q))\right].\end{aligned}$$

Using the definition of $\rho$,

$$\begin{aligned}\mathcal{L}_\rho(\theta) &= \mathbb{E}_{q\sim\mathcal{D}}[\sum_{s,a} \pi_\theta(s,a|q) \sum_{s'} \pi_\theta(s'|q,a)\, \pi_\theta(a^*(q)|s',q)] \\ &= \mathbb{E}_{q\sim\mathcal{D}}[\sum_{s'} \pi_\theta(a^*(q)|s',q) \sum_{s,a} \pi_\theta(s,a|q)\, \pi_\theta(s'|q,a)].\end{aligned}$$

For fixed $q$, we have

$$\begin{aligned}&\sum_{s,a} \pi_\theta(s,a|q)\, \pi_\theta(s'|q,a) \\ &= \sum_{a} \pi_\theta(a|q)\, \pi_\theta(s'|q,a) \\ &= \pi_\theta(s'|q),\end{aligned}$$

where the first equality marginalizes out $s$ and the second follows from the law of total probability.

Therefore,

$$\begin{aligned}\mathcal{L}_\rho(\theta) &= \mathbb{E}_{q\sim\mathcal{D}}\left[\sum_{s'} \pi_\theta(s'|q)\, \pi_\theta(a^*(q)|s',q)\right] \\ &= \mathbb{E}_{q\sim\mathcal{D}}\left[\sum_{s',a} \pi_\theta(s',a|q)\, \mathbb{I}(a = a^*(q))\right] \\ &= \mathbb{E}_{q\sim\mathcal{D},(s,a)\sim\pi_\theta(\cdot|q)}[\mathbb{I}(a = a^*(q))].\end{aligned}$$

This shows that the expected CER objective is equivalent in value to the exact-match objective. □

### A.3 EXPERIMENTS

The additional experiments in this section serve the following purposes:

- **Length Normalization:** Analyze potential length bias in likelihood-based rewards and examine whether length normalization can alleviate this issue, providing a simple way to improve the robustness of CER against answer-length variation.
- **Evaluation on Open-ended Task:** Evaluate whether CER generalizes to open-ended writing tasks with diverse surface forms and semantic variations, demonstrating its applicability beyond exact-match and rule-verifiable benchmarks.
- **Failure Cases:** Identify representative failure modes of likelihood-based reward estimation and examine the complementary role of rule-based verification through CER+Rule, showing how explicit verification can mitigate model-likelihood bias while preserving CER's ability to capture soft answer similarity.
- **Standard Deviations:** Assess the training stability of CER by measuring performance variation across independent runs with different random seeds, providing evidence that the observed improvements are consistent across runs rather than primarily caused by random initialization or training fluctuations.
- **Hyperparameter Settings:** Provide detailed training and evaluation configurations, improving the transparency and reproducibility of our experimental results.

**Length Normalization** We also analyze the potential length bias of likelihood-based rewards. CER can be integrated with length normalization by downscaling the log-reward according to the answer length (Kwiatkowski et al., 2026). We evaluate CER with and without length normalization, and the results are shown in Table 4.

The two variants achieve comparable performance, with length normalization slightly improving the average score from 49.0 to 49.3. This result indicating that length normalization can alleviate potential length bias in CER.

Table 4: Comparison of CER with and without length normalization.

| Method | w/o Normalization | w/ Normalization |
|---|---|---|
| MATH500 | 82.6 | 83.8 |
| AMC23 | 66.7 | 67.7 |
| AIME24 | 22.8 | 22.4 |
| AIME25 | 21.7 | 21.4 |
| MMLU-Pro | 64.7 | 64.4 |
| SuperGPQA | 36.0 | 35.9 |
| Average | 49.0 | 49.3 |

**Evaluation on Open-ended Task** To further evaluate the advantage of CER in handling diverse free-form responses, we test CER on WritingBench (Wu et al., 2026), a benchmark for assessing LLM writing ability on diverse open-ended tasks, including creative, informative, persuasive, and technical writing. Unlike exact-match benchmarks, WritingBench adopts an LLM-based, criteria-aware evaluation framework: task-specific criteria are first generated, and model responses are then scored according to these criteria.

As shown in Table 5, CER achieves the best performance among all compared methods on WritingBench, demonstrating its ability to evaluate free-form responses with diverse surface forms and semantic variations. This result complements the MMLU-Pro and SuperGPQA evaluations by directly validating the advantage of CER in open-ended generation scenarios.

**Failure Cases** We acknowledge that CER relies on model likelihood estimation and may therefore be affected by the model's calibration and inherent biases. In particular, likelihood-based similarity can be sensitive to surface-form differences and may not always faithfully reflect semantic correctness.

Table 5: Performance on WritingBench. Scores range from 0 to 10, with higher scores indicating better performance.

| Method | Score (0–10) |
|---|---|
| Exact-Match | 6.56 |
| Rule-based | 7.27 |
| VeriFree | 7.14 |
| RLPR | 7.21 |
| Nover | 7.34 |
| General-verifier | 7.32 |
| CER | 7.65 |
| CER+Rule | 7.73 |

For example, CER assigns an extremely low reward ($2.6 \times 10^{-5}$) to '986737.76" when the reference answer is "$986,737.76$", despite the two answers being numerically equivalent. Conversely, CER assigns a relatively high reward (0.45) to "13.54" when the reference answer is "13.54%", although the two answers have different semantic meanings. These cases illustrate that model likelihood alone may incorrectly penalize semantically valid answers or assign excessive reward to incorrect ones.

To mitigate these limitations, we further introduce CER+Rule, a hybrid variant that combines the complementary strengths of rule-based verification and CER. Rule-based verification provides an explicit correctness signal when reliable rules can be constructed, while CER provides a softer, likelihood-based signal that can capture answer similarity beyond exact rule matching. By jointly leveraging these two signals, CER+Rule reduces the influence of model likelihood bias while retaining CER's ability to provide informative rewards when rule-based verification is insufficient. Empirically, CER+Rule consistently achieves better performance than either rule-based verification or CER alone, demonstrating that the two signals are complementary and that combining them provides a more robust reward for answer evaluation.

**Standard deviations** We conducted three independent training runs with different random seeds and report the corresponding standard deviations for all methods in the revised manuscript.

As shown in Table 6, performance variations across runs are generally small, indicating that the improvements of our method are not primarily attributable to random initialization or training fluctuations, but are consistent across independent runs.

Table 6: Standard deviations over three independent training runs with different random seeds.

| Method | MATH500 | AMC23 | AIME2024 | AIME2025 | MMLU-Pro | SuperGPQA | Average |
|---|---|---|---|---|---|---|---|
| Exact-Match | 0.18 | 0.09 | 0.16 | 0.07 | 0.21 | 0.13 | 0.14 |
| Rule-based | 0.15 | 0.14 | 0.12 | 0.09 | 0.13 | 0.14 | 0.12 |
| VeriFree | 0.11 | 0.19 | 0.15 | 0.13 | 0.08 | 0.22 | 0.17 |
| RLPR | 0.19 | 0.26 | 0.21 | 0.17 | 0.25 | 0.22 | 0.22 |
| Nover | 0.22 | 0.16 | 0.20 | 0.24 | 0.15 | 0.18 | 0.19 |
| General-verifier | 0.08 | 0.12 | 0.18 | 0.16 | 0.10 | 0.25 | 0.13 |
| CER | 0.19 | 0.12 | 0.11 | 0.12 | 0.18 | 0.15 | 0.16 |
| CER+Rule | 0.10 | 0.08 | 0.21 | 0.15 | 0.12 | 0.09 | 0.09 |

**Hyperparameter settings** We set the batch size of questions to 32, the number of solutions $N$ to 16, $M$ in Eq. (5) to 16, the learning rate to $10^{-6}$, epoch to 1 for WebInstruct dataset and epoch to 5 for MATH-7.5K dataset. For training, we use temperature $= 1.0$ and top-$p = 1.0$, while for evaluation we use temperature $= 0.6$, top-$p = 0.95$ and top-$k = 20$. The maximum question length is set to $2048$ tokens, and the maximum output length is set to $4096$ tokens for training and $8192$ for evaluation. We utilize RLOO (Kool et al., 2019) as the optimization method.