SERL: Self-Examining Reinforcement Learning on Open-Domain

Weixuan Ou[1*], Yanzhao Zheng[2*], Shuoshuo Sun[2], Wei Zhang[2], Baohua Dong[2†], Hangcheng Zhu[2], Ruohui Huang[2], Gang Yu[2], Pengwei Yan[1], Yifan Qiao[2]

[1]Zhejiang University, Hangzhou, China
[2]Alibaba Group, Hangzhou, China
{ouweixuan, yanpw, pureidea.f}@zju.edu.cn@stu.xjtu.edu.cn
{zhengyanzhao.zyz, sunshuoshuo.sss, zw443021, baohua.dbh, linran.lr09, wentong, ruohai}@alibaba-inc.com

Abstract

Reinforcement Learning (RL) has been shown to improve the capabilities of large language models (LLMs). However, applying RL to open-domain tasks faces two key challenges: (1) the inherent subjectivity of these tasks prevents the verifiable rewards as required by Reinforcement Learning with Verifiable Rewards (RLVR); (2) Reinforcement Learning from Human Feedback (RLHF) relies on external reward mechanisms. To overcome these limitations, we propose **S**elf-**E**xamining **R**einforcement **L**earning (**SERL**), a novel self-improving framework where the LLM serves as both Actor and Judge. SERL introduces two synergistic reward mechanisms without any external signals. On the one hand, to improve the Actor's capability, we derive rewards from Copeland-style pairwise comparison judgments across a group of generated responses. On the other hand, a self-consistency reward that encourages coherent judgments is proposed to enhance the Judge's reliability. This refinement strengthens the Judge, consequently generating a more robust training signal for the Actor. Experiments show that our method outperforms existing self-improvement training methods. SERL improves the LC win rate of Qwen3-8B on AlpacaEval 2.0 from 52.37% to **59.90%**. To the best of our knowledge, our method achieves state-of-the-art performance among self-improving approaches. Furthermore, it achieves a performance comparable to significantly larger models like Qwen3-32B, demonstrating superior effectiveness and robustness on open-domain tasks.

Code — https://github.com/AlwaysOu/SERL

1 Introduction

Reinforcement Learning (RL) has been demonstrated as an effective post-training approach that significantly improves the generation and reasoning capabilities of large language models (Jaech et al. 2024; Guo et al. 2025). In particular, Reinforcement Learning with Verifiable Rewards (RLVR) has shown remarkable efficacy in specialized tasks such as mathematical reasoning (Hu et al. 2025; Yu et al. 2025a) and code generation (Luo et al. 2025). However, improving model performance on a broad range of open-domain tasks through RL still faces two major challenges.

Unverifiable answers: Open-domain tasks, including open writing and summarization, typically lack definitive correct answers, which are indispensable factors for RLVR. Ma et al. (2025) try to solve this chanllenge by training a verification model to evaluate the consistency between policy answers and reference answers, while (Zhou et al. 2025; Yu et al. 2025b) leverage the policy itself to compute the joint distribution between the reasoning process and reference answers as rewards. These works only focus on objective question-answering tasks such as MMLU (Wang et al. 2024), GPQA (Rein et al. 2024), TheoremQA (Chen et al. 2023), etc., while neglecting more open-ended task types such as summarization, open writing, and even general-domain open-QA. In practice, these tasks usually lack reference answers or supply only low-quality ones.

Reliance on external reward mechanisms: Earlier methodologies, such as Reinforcement Learning from Human Feedback (RLHF) (Stiennon et al. 2020) and Reinforcement Learning from AI Feedback (RLAIF) (Lee et al. 2023), have demonstrated effectiveness in improving model capabilities across general domains by leveraging feedback from human annotators or AI-based evaluators. However, these RL methods also face significant limitations: they require extensive data annotation or dedicated reward models, leading to scalability challenges and additional computational overhead (Gao, Schulman, and Hilton 2023). Yuan et al. (2024); Wu et al. (2024) introduce an offline self-improving method, in which the model scores each of its own outputs individually and then constructs preference data for DPO training in each iteration. These approaches require supervised cold-start fine-tuning, and their point-wise scoring depends on well-crafted standards, limiting their generalizability across diverse tasks.

To address these limitations, we propose SERL, a novel self-examining reinforcement learning framework designed specifically to enhance LLM generation capabilities in open-domain scenarios without relying on any external supervision signals or reward mechanisms. Our framework introduces a self-examining mechanism in which the model alternately assumes the roles of Actor and Judge, jointly optimizing its generation and evaluation capabilities. During training, the model samples diverse responses for each in-

[*]These authors contributed equally.
[†]Corresponding author
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

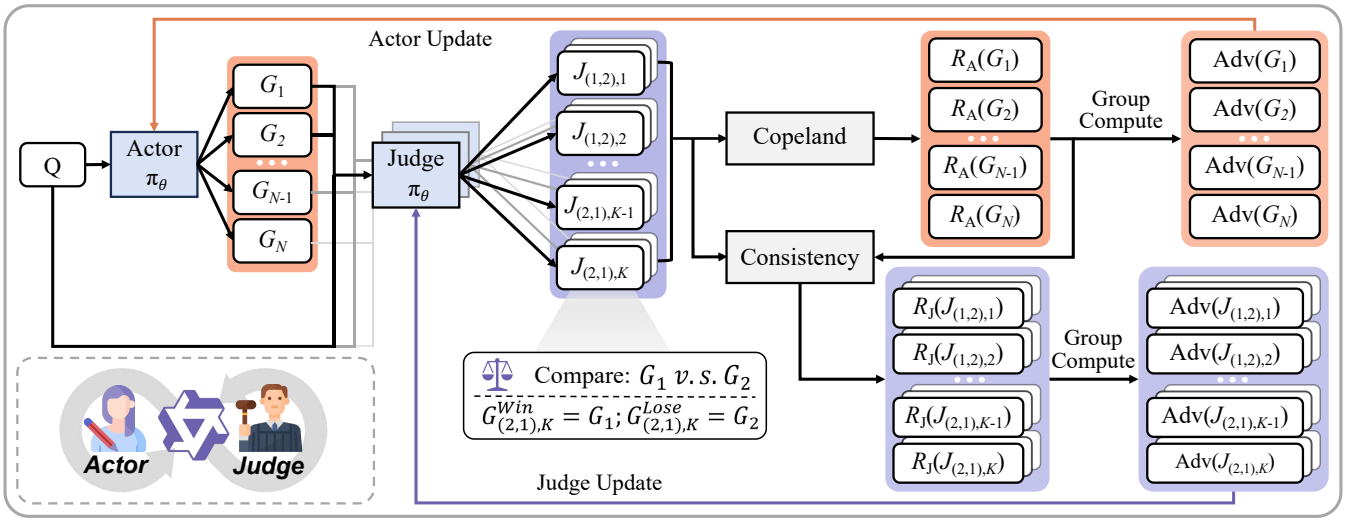

Figure 1: Overview of SERL. Given a instruction, the Actor first samples a group of responses. The Judge samples pairwise comparison judgments between response pairs. The judgments are aggregated via Copeland method to yield the Reward for Actor. Next, the consistency between the judgments and the ranking implied by the Reward for Actor is computed to generate the Reward for Judge. This process jointly enhances generation ability and comparative evaluation ability.

Copeland method

	G_1	G_2	G_3	G_4	Win Times	Win Rate
G_1	–	Win	Win	Lose	2	2/3(66.67%)
G_2	Lose	–	Win	Win	2	2/3(66.67%)
G_3	Lose	Lose	–	Win	1	1/3(33.33%)
G_4	Win	Lose	Lose	–	1	1/3(33.33%)

Figure 2: Illustration of the Copeland method.

put and then evaluates these responses by itself. Specifically, we introduce Copeland-style judgments as shown in Figure 2, in which model conducts pairwise comparisons between responses and subsequently ranks them according to their respective win rates within one group, establishing ranking rewards for generations and intrinsic consistency rewards for evaluations. SERL leverages the model's intrinsic evaluation capabilities while ensuring co-evolution of both generation and evaluation abilities via online learning, thus removing reliance on external reward signals.

Experiments on multiple open-domain tasks demonstrate the efficacy of SERL. Based on Qwen3-8B, our model achieves a notable performance increase on diverse open-domain tasks. Only a few dozen steps of training elevate the LC win rate on AlpacaEval 2.0 from 52.37% to 59.90%. Furthermore, extensive experiments across summarization and open writing tasks show that SERL consistently outperforms other mainstream self-improving methods, and delivering performance competitive with significantly large models like Qwen3-32B.

In summary, our main contributions are as follows:

1. We introduce SERL, a novel self-examining reinforcement learning framework that enables large language models to jointly optimize their generation and evaluation abilities without any external supervision signals, verifier models, or human annotations, thereby inherently enjoying scalability and flexibility on open-domain tasks.
2. Extensive experiments show that SERL achieves remarkable training efficiency and performance improvements on diverse open-domain tasks, including summarization, open writing, and General-QA. With only dozens of training steps, SERL boosts Qwen3-8B's LC win rate on AlpacaEval 2.0 by 7.53%, and delivers competitive or superior performance to much larger models such as Qwen3-32B. To the best of our knowledge, our method achieves state-of-the-art performance among self-improving approaches.

2 Related Works

RLVR Works on Reinforcement Learning with Verifiable Rewards (RLVR) have achieved improvements in the capabilities of large models. Representative works include CodeRL (Le et al. 2022), RLTF (Liu et al. 2023), and PPOCoder (Shojaee et al. 2023) in programming. Recently, Group Relative Policy Optimization (GRPO) (Shao et al. 2024) improves sample efficiency by estimating advantages within a sampled group without a learned value network. GRPO and follow-up works such as DAPO (Yu et al. 2025a) and VAPO (Yue et al. 2025) demonstrate continued improvements in math. However, they cannot be directly applied to open-domain settings due to the lack of verifiable answers for these tasks. SERL preserves GRPO's group-wise efficiency while computing differential rewards through pairwise comparison judgments, thereby overcoming the reward-design bottleneck on open-domain tasks where verifiable rewards are unavailable.

RLHF and RLAIF Reinforcement Learning from Human Feedback (RLHF) involves training a reward model on human pairwise comparisons and subsequently optimizing the policy with algorithms like PPO. This approach has achieved strong alignment in prominent models such as InstructGPT (Ouyang et al. 2022), ChatGPT/GPT-4 (Achiam et al. 2023), and Llama-2-chat (Touvron et al. 2023). To improve efficiency, variants like Direct Preference Optimization (DPO) (Rafailov et al. 2023), KTO (Ethayarajh et al. 2024), and Online-DPO (Lanchantin et al. 2025) simplify or accelerate preference learning, yet still depend on human labels. Reinforcement Learning from AI Feedback (RLAIF) (Bai et al. 2022a) was introduced to alleviate the costly process of human annotation, which uses the LLM itself as the labeler. This includes methods like Constitutional AI (Bai et al. 2022b), which apply fixed principles for self-critique. As noted in Self-Rewarding (Yuan et al. 2024), employing the Actor as a Judge to provide feedback has been shown to enhance the Actor's capabilities. However, it lacks optimization for Judge. In contrast, Meta-Rewarding (Wu et al. 2024) jointly optimize both the Actor and the Judge, but it relies on off-policy learning paradigms. We propose SERL, an on-policy training framework that jointly optimizes the model's capabilities as both Actor and Judge, independent of external signals. In SERL, the same model simultaneously acts as Actor to generate multiple reasoning traces and as Judge to compare them, thereby obviating the need for external human or proxy preference labels.

3 Method

SERL is a self-improving framework designed to comprehensively improve a model's performance on open-domain tasks. Its methodology is centered on the model assuming two primary roles: Actor and Judge. As the Actor, it generates responses to given prompts. As the Judge, it compares the relative strengths and weaknesses among its own generated responses. The rewards for Actor and Judge are constructed without reliance on any external signals, synergistically improving the model's generation and evaluation capability.

SERL's training pipeline (Figure 1) is as follows:

- **Generation.** For each prompt, the Actor stochastically samples a set of diverse responses. This step creates the raw material for examination and refinement.
- **Examination.** For each response pair, the Judge employs an "LLM-as-a-Judge" prompting strategy to sample a set of pairwise comparison judgments.
- **Rewards.** For a set of pairwise comparison judgments, Reward for Actor ($\mathcal{R}_{\text{A}}$) is calculated as the win rate of each response across all pairwise comparisons. $\mathcal{R}_{\text{A}}$ for each response can be interpreted as its rank among the generated responses. Reward for Judge ($\mathcal{R}_{\text{J}}$) is calculated as the consistency between this ranking and each individual pairwise comparison judgment.

This dual-reward mechanism is analogous to how humans approach complex problems by attempting multiple solutions, comparing the effectiveness of these approaches, and ultimately becoming proficient in solving such problems. Without relying on any external signals, this mechanism refines model's evaluation capability (as Judge), which in turn provides a more robust reward to enhance model's generation capability (as Actor).

3.1 Generation

At step t, given a set of questions $Q = \{q\}$, we sample a group of N individual responses for each input q from the old Actor π_{old}: $\{G_n\}_{n=1}^{N} \sim \pi_{\text{old}}$.

3.2 Examination

We deliberately choose comparison as the core evaluation mechanism because of the widely recognized principle that for both humans and LLMs (Ouyang et al. 2022; Zheng et al. 2023), making relative judgments (i.e., "Is response A better than response B?") is a more tractable and reliable task than assigning absolute scores (e.g., "Rate response A on a scale of 1 to 10"). For open-domain tasks that lack verifiable answers and explicitly defined evaluation criteria, framing the evaluation as a series of forced-choice preference selections enables us to elicit more stable, consistent, and meaningful feedback, which is crucial for effectively guiding the iterative refinement of our models.

To aggregate these individual preference judgments into a coherent global ranking, we employ the Copeland method (Copeland 1951; Dwork et al. 2001). As illustrated in Figure 2, this method treats each response as a "candidate" and tallies the results of all possible pairwise matchups. This approach is not only intuitive and robust but also effectively resolves potential preference cycles, providing a clear and defensible final ordering.

At step t, the Judge evaluates comparisons to determine the preferred response between G_i and G_j, where $i, j \in \{1, ..., N\}$, $i \neq j$. To enhance the reliability of this process, we employ a multi-sample evaluation strategy. For each pair (G_i, G_j), we sample K independent judgments $J_{(i,j),k}$ from the old Judge π_{old}: $\{J_{(i,j),k}\} \sim \pi_{\text{old}}$. We define $G^{\text{Win}}_{(i,j),k}$ and $G^{\text{Lose}}_{(i,j),k}$ as the winning and losing responses, in the comparison between $\{G_i, G_j\}$, based on the k-th judgment $J_{(i,j),k}$.

Position Bias Mitigation Mechanism Some works have found that "LLM-as-a-Judge" exhibits position bias, the tendency of judgments to disproportionately favor responses based on their placement within an input list rather than their intrinsic merit (Shi et al. 2024). To mitigate positional bias, we introduce the Position Bias Mitigation Mechanism (PBMM) that swaps the positions in half of the K pairwise comparisons: $K//2$ prompts constructed as (q, G_i, G_j), and $K//2$ constructed as (q, G_j, G_i).

3.3 Rewards

Reward for Actor We introduce the metric $\mathcal{R}_{\text{A}}$ to enhance model's generation capability. Specifically, $\mathcal{R}_{\text{A}}$ for each response G_n is defined as its win rate across all pairwise comparisons with other candidate responses. This metric provides a clear, direct quantification of the relative quality of G_n compared to other responses, thus offering an explicit

and informative learning signal to guide and optimize the Actor's generation behavior. $\mathcal{R}_{\text{A}}$ is calculated as follows:

$$\mathcal{R}_{\text{A}}(G_n) = \frac{\sum_{i \neq j,k} \mathbf{1}(G_n = G^{\text{Win}}_{(i,j),k})}{M \times K} \tag{1}$$

where M is calculated as $\binom{N}{2}$, the number of pairwise combinations involving $\{G_n\}$.

Length Control Module Length-bias means the model's tendency to favor longer responses (Dubois et al. 2024; Park et al. 2024). To ensure the evaluations are conducted under similar output length conditions, we introduce a Length Control Module (LCM). Accordingly, the $\mathcal{R}_{\text{A}}$ is further modified as follows:

$$\mathcal{R}_{\text{A}}(G_n) = \frac{\sum_{i \neq j,k} \beta \left(G_n = G^{\text{Win}}_{(i,j),k}\right)}{M \times K} \tag{2}$$

$$\beta = \frac{|(G^{\text{Lose}}_{(i,j),k})|}{|(G^{\text{Win}}_{(i,j),k})|} \tag{3}$$

$$\text{s.t. } (1-\alpha) < \frac{|(G^{\text{Win}}_{(i,j),k})|}{|(G^{\text{Lose}}_{(i,j),k})|} < (1+\alpha)$$

The hyperparameter α is used to restrict valid comparisons to only those pairwise responses where lengths are sufficiently similar. Specifically, a smaller value of α imposes a stricter requirement for the lengths of the compared responses to be closer. In our setup, $\alpha = 0.2$. The ratio β serves to balance the influence of response length on the win-loss determination among the valid comparisons, assigning a greater reward to shorter winning responses.

Reward for Judge $\mathcal{R}_{\text{J}}$ is proposed to improve model's evaluation capabilities. We define $\mathcal{R}_{\text{J}}$ to measure the consistency between the global ranking of responses and each pairwise comparison judgement. A judgment is considered consistent if the selected winner in a pairwise comparison ranks higher than the loser in the global ranking. Higher consistency indicates the model's evaluative confidence, offering clear signals for reliable and coherent judgment decisions. $\mathcal{R}_{\text{J}}$ is described as follows:

$$\mathcal{R}_{\text{J}}(J_{(i,j),k}) = \text{sign}\left(\mathcal{R}_{\text{A}}(G^{\text{Win}}_{(i,j),k}) - \mathcal{R}_{\text{A}}(G^{\text{Lose}}_{(i,j),k})\right) \tag{4}$$

where the sign function is defined as:

$$\text{sign}(x) = \frac{x}{|x|} \quad (x \neq 0), \quad \text{sign}(0) = 0 \tag{5}$$

3.4 Online Optimization

The model is optimized on a mixed set of responses and judgments. The SERL training objective, adapted from GRPO, is formulated with a surrogate function Surr. as:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\substack{q \sim Q, \\ G \sim \pi_{\text{old}}}} \left[\frac{1}{N} \sum_{n=1}^{N} \text{Surr.}(\text{Actor}, G_n, n; \theta) \right] \tag{6}$$

$$\mathcal{J}_{\text{SERL}}(\theta) = \mathbb{E}_{\substack{q \sim Q, \\ G, J \sim \pi_{\text{old}}}} \Bigg[\frac{1}{N} \sum_{n=1}^{N} \text{Surr.}(\text{Actor}, G_n, n; \theta) + \frac{1}{M \times K} \sum_{i \neq j} \sum_{k=1}^{K//2} \text{Surr.}(\text{Judge}, J_{(i,j),k}, [(i,j),k]; \theta) \Bigg] \tag{7}$$

where the objective function Surr. for a single sampled output, the advantages calculated by normalizing the group-level rewards $\hat{A}^{\text{Actor}}_{n,t}$, $\hat{A}^{\text{Judge}}_{(i,j),k,t}$ for all tokens in a response, and the token-level probability ratios $r^{\text{Actor}}_{n,t}(\theta)$, $r^{\text{Judge}}_{(i,j),k,t}(\theta)$, are defined as:

$$\text{Surr.}(R, o, i;\ \theta) = \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[r^{R}_{i,t}(\theta)\, \hat{A}^{R}_{i,t}, \text{clip}\Big(r^{R}_{i,t}, 1-\epsilon,\ 1+\epsilon\Big)\, \hat{A}^{R}_{i,t} \right] \tag{8}$$

$$\hat{A}^{\text{Actor}}_{n,t} = \frac{\mathcal{R}_{\text{A}} - \text{mean}(\{\mathcal{R}_{\text{A}}\}^{N})}{\text{std}(\{\mathcal{R}_{\text{A}}\}^{N})} \tag{9}$$

$$\hat{A}^{\text{Judge}}_{(i,j),k,t} = \frac{\mathcal{R}_{\text{J}} - \text{mean}(\{\mathcal{R}_{\text{J}}\}^{M \times K})}{\text{std}(\{\mathcal{R}_{\text{J}}\}^{M \times K})} \tag{10}$$

$$r^{\text{Actor}}_{n,t}(\theta) = \frac{\pi_{\theta}(G_{n,t} \mid q, G_{n,<t})}{\pi_{\theta_{\text{old}}}(G_{n,t} \mid q, G_{n,<t})} \tag{11}$$

$$r^{\text{Judge}}_{(i,j),k,t}(\theta) = \frac{\pi_{\theta}(J_{(i,j),k,t} \mid q, J_{(i,j),k,<t})}{\pi_{\theta_{\text{old}}}(J_{(i,j),k,t} \mid q, J_{(i,j),k,<t})} \tag{12}$$

In GRPO, the KL penalty term is used to regulate the divergence between the online policy and the frozen reference policy. However, recent work suggests that during the training of long-CoT reasoning model, the model distribution can diverge significantly from the initial model (Yu et al. 2025a). Therefore, we exclude the KL term in SERL.

4 Experiments

4.1 Experimental Setup

Datasets To comprehensively evaluate the effectiveness and generalization capability of our proposed method SERL, we conduct experiments on three tasks including summarization, open writing, and general QA.

- **Summarization** CNN-DM (Hermann et al. 2015): A standard dataset for abstractive text summarization, consisting of news articles and their corresponding human-written summaries.
- **Open Writing** writingprompts (Fan, Lewis, and Dauphin 2018): A large-scale dataset containing open writing prompts and human-written stories.
- **General QA** (1) AlpacaEval 2.0 (Dubois et al. 2024): A recently developed benchmark that includes a wide range of general QA tasks spanning multiple domains. This allows us to test the general applicability of our method in handling diverse questions and producing accurate and

Method1	Method2	Summarization			Open Writing		
		Win rate	ΔWin rate	Len	Win rate	ΔWin rate	Len
SERL(Ours)	Online-DPO	55.17%	+10.33%	1190 : 1797	50.50%	+1.00%	1654 : 1767
SERL(Ours)	Self-Rewarding	59.50%	+19.00%	1190 : 922	55.17%	+10.33%	1654 : 1887
SERL(Ours)	Meta-Rewarding	59.17%	+18.33%	1190 : 1441	56.67%	+13.33%	1654 : 2189
SERL(Ours)	RLSC	86.17%	+72.33%	1190 : 1322	-	-	-
SERL(Ours)	GRPO(ROUGE-L)	99.17%	+98.33%	1190 : 248	-	-	-

Table 1: Evaluation results of SERL against other methods on summarization and open writing tasks. ΔWin rate is the difference in win rates between two methods.

Models	General QA (AlpacaEval 2.0)		
	LC win rate	Win rate	Len
Online-DPO	54.07%	59.74%	3429
Self-Rewarding	51.29%	53.69%	3074
Meta-Rewarding	54.73%	55.93%	3081
RLSC	52.11%	51.81%	2060
SERL(Ours)	**59.90%**	**69.88%**	3017

Table 2: Evaluation results of SERL and other methods on general QA task.

coherent responses. (2) Ultrafeedback (Ye et al. 2023): A large-scale, diverse, and fine-grained preference dataset. We select this dataset as the training dataset for the general QA task because it contains diverse prompts from datasets such as TruthfulQA, FalseQA, Evol-Instruct, UltraChat, and ShareGPT.

As SERL is independent of external reward signals, we use only the prompts from these datasets. The dataset statistics are presented in Appendix C.

Baselines

- **Self-Rewarding** (Yuan et al. 2024) Employs the model itself as a reward estimator to iteratively refine its own outputs, but lacks formal optimization for the reward estimator.
- **Meta-rewarding** (Wu et al. 2024) Jointly optimizes the Actor and Judge via off-policy learning, decoupling feedback generation from policy updates.
- **Online-DPO** (Lanchantin et al. 2025) Recent work has found that Online-DPO achieves performance comparable to GRPO. In self-improving setting, we adopt the policy model itself as a reward model to compare sampled pairwise responses.
- **RLSC** (Li et al. 2025) This method introduces confidence-aware reinforcement learning, also without reliance on any external signals, to balance exploration and exploitation, showing improvement in math.
- **GRPO** (Shao et al. 2024) In summarization task, we employ the ROUGE-L score (Lin and Hovy 2003), computed by comparing the output against the reference summary from the dataset, as the reward for GRPO.

To ensure a fair comparison, we remove the Supervised Fine-Tuning (SFT) on the Instruction Fine-Tuning (IFT) and Evaluation Fine-Tuning (EFT) data for the Self-Rewarding and Meta-Rewarding methods, which would introduce an external training signal.

Implementation Details and Evaluation Metrics We adopt the Qwen3-8B as base model for all methods. All baselines use their authors' recommended hyperparameters. For SERL, we set the sample number of Actor N as 4, the sample number of Judge K as 4, topP=1.0, topK=50, and temperature = 0.9 during generation and examination. Other parameters like epoch and learning rate are set differently for various tasks. Other details are provided in Appendix C.

For both summarization and open writing tasks, we select GPT-4o as the evaluator to compare two outputs—either summaries or story continuations—generated from two methods on the same instruction, selecting the better output. To reduce positional bias, we compared the outputs from two methods twice. For the second comparison, we swapped the order of the two responses. If both pairwise comparisons identify the same model's output as the winner, then give a WIN for that model and a LOSE to the other; otherwise, the result is considered a TIE. For these two tasks, we define the win rate as follows:

$$\text{Win rate} = \frac{2 \times \text{WIN} + \text{TIE}}{2 \times \text{WIN} + \text{TIE} + 2 \times \text{LOSE}} \tag{5}$$

For the general QA task, we evaluate on AlpacaEval 2.0, reporting the win rate and Length-controlled (LC) win rate.

4.2 Comparisons with Self-improving Methods

The overall performance of SERL and baseline methods is presented in Tables 1 and 2. Based on the results, we have the following key observations: (1) Our method achieves superior performance across all tasks. It outperforms baselines on summarization and open writing tasks. Furthermore, it achieves the highest LC win rate (59.90%) and win rate (69.88%) in AlpacaEval 2.0. (2) The performance of RLSC is inconsistent with its effectiveness in mathematical tasks, which means this method is not well suited for open-domain tasks because its objective of mode sharpening conflicts with the goal of fostering diverse and creative responses. (3) In the summarization task, GRPO, which uses ROUGE-L as reward, shows a significant win-rate gap compared to SERL. This is primarily due to the unreliable quality of human annotations on open-domain tasks or reward hacking. SERL

Model1	Model2	Summarization			Open Writing		
		Win rate	Win rate↑	Len	Win rate	Win rate↑	Len
Qwen3-8B(base)	Qwen3-32B	37.33%	-	1066 : 1109	33.00%	-	1369 : 1506
Qwen3-8B(base)	R1-Distill-Qwen-32B	63.50%	-	1066 : 922	78.50%	-	1369 : 1754
Qwen3-8B(base)	R1-Distill-Llama-70B	63.67%	-	1066 : 1012	71.33%	-	1369 : 2023
Qwen3-8B(base)	Claude 3.5 Sonnet	48.33%	-	1066 : 1064	81.33%	-	1369 : 1632
Qwen3-8B(base)	GPT-4o-0513	55.33%	-	1066 : 941	63.50%	-	1369 : 1412
SERL(Ours)	Qwen3-8B(base)	62.83%	+12.83%	1190 : 1066	61.50%	+11.50%	1654 : 1369
SERL(Ours)	Qwen3-32B	52.67%	+15.33%	1190 : 1109	46.67%	+13.67%	1654 : 1506
SERL(Ours)	R1-Distill-Qwen-32B	72.17%	+8.67%	1190 : 922	88.67%	+10.17%	1654 : 1754
SERL(Ours)	R1-Distill-Llama-70B	71.00%	+7.33%	1190 : 1012	84.00%	+12.67%	1654 : 2023
SERL(Ours)	Claude 3.5 Sonnet	56.67%	+8.33%	1190 : 1064	91.00%	+9.67%	1654 : 1632
SERL(Ours)	GPT-4o-0513	65.50%	+10.17%	1190 : 941	73.67%	+10.17%	1654 : 1412

Table 3: Evaluation results of SERL against other general-purpose LLMs on summarization and open writing tasks. Win rate↑ is the relative win rate improvement, which is calculated as the difference between the win rate of the model trained with our method against other models and the win rate of the base model against the same models.

Models	General QA (AlpacaEval 2.0)		
	LC win rate	Win rate	Len
Qwen3-8B	52.37%	55.07%	3100
Qwen3-32B	**62.16%**	66.47%	3034
R1-Distill-Qwen-32B	41.76%	37.79%	1817
R1-Distill-Llama-70B	54.06%	44.51%	1617
Claude 3.5 Sonnet	57.77%	44.70%	1487
GPT-4o-0513	57.46%	51.33%	1873
SERL(Ours)	59.90%	**69.88%**	3017

Table 4: Evaluation results of SERL and other general-purpose LLMs on general QA task.

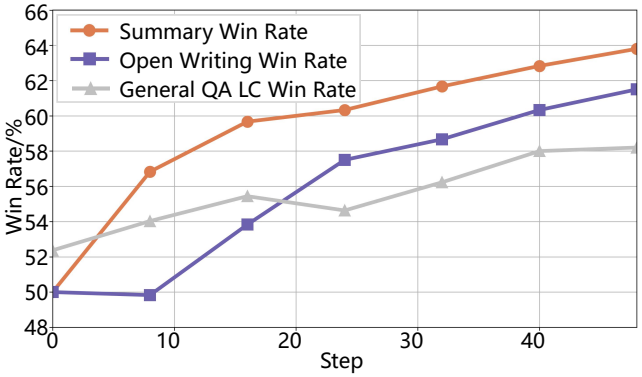

Figure 3: The win rate against Qwen3-8B on summarization, open writing, and general QA with SERL training step.

avoids reliance on manual annotations, instead enhancing model performance through self generation and self evaluation. These results empirically validate the effectiveness of SERL.

4.3 Improvement of Win Rate in SERL Iterations

Figure 3 shows the win rates against the base model Qwen3-8B on three tasks for SERL over its training iterations. Notably, the model achieves an average performance gain of 10.33% within just 48 training steps. In general, we observe substantial improvements on each task. Considering that our model has only 8B parameters and is trained without external signals, this is a remarkable result. Moreover, we observe similar improvements on the smaller model Qwen3-1.7B on these three tasks, as detailed in Appendix B. These findings indicate that our method effectively enhances the capabilities of the initial model in these tasks.

4.4 Comparisons with General-purpose LLMs

The comparative results between SERL and general-purpose LLMs are shown in Table 3 and 4. In summarization and open writing tasks, compared with the Qwen3-8B, the win rates of the model trained with SERL are 62.83% and 61.50%, respectively. In the general QA task, SERL improves the LC win rate and the win rate of Qwen3-8B from 52.37% and 55.07% of the base model to 59.90% and 69.88%, respectively, marking gains of 7.53% and 14.81%.

Compared with the stronger model Qwen3-32B within the same model series, the model trained based on Qwen3-8B achieves comparable performance: it slightly surpasses Qwen3-32B by 2.67% on the summarization task, shows only a 3.33% disadvantage on the open writing task, and achieves very close results on the general QA task, trailing by 2.26% in LC win rate but leading by 3.41% in win rate.

Comparisons with other models show that the model trained with SERL significantly outperforms Deepseek-R1-Distill-Qwen-32B, Claude 3.5 Sonnet, and GPT-4o. SERL achieves consistent Win rate↑ with maximum gains of 11.83% in summarization task and 10.17% in open writing task. In the general QA task, both the win rate and the LC win rate exceed those of the competing models.

4.5 Consistency Verification of Evaluation

To verify the consistency of using LLMs as evaluators for result evaluation, we employed different evaluator models

Ablation Variants	Summarization		Open Writing		General QA (AlpacaEval 2.0)			
	Win rate	ΔWin rate	Win rate	ΔWin rate	LC Win rate	Win rate	ΔLC win rate	ΔWin rate
SERL wo $\mathcal{R}_J$	45.33%	-9.34%	44.67%	-10.66%	54.50%	65.47%	-5.40%	-4.41%
SERL wo $\mathcal{R}_A$	32.33%	-35.34%	39.83%	-20.34%	51.47%	54.95%	-8.43%	-14.93%
SERL wo PBMM	42.50%	-15.00%	48.67%	-2.66%	54.01%	63.33%	-5.89%	-6.55%

Table 5: Ablation experiment results. Comparison between the ablation variants with the complete method on summarization and open writing tasks and Performance of variants on general QA task. ΔWin rate indicates the difference in win rates.

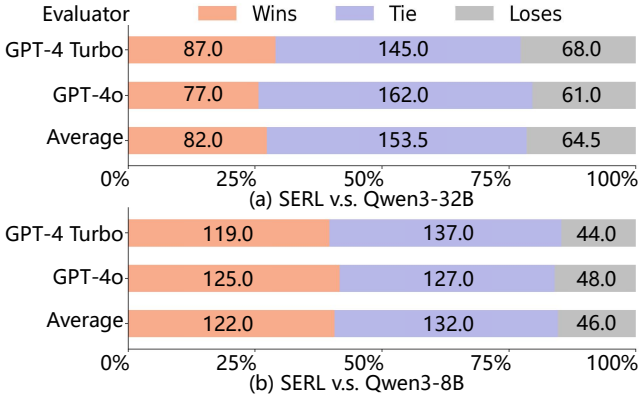

Figure 4: Consistency of evaluation results across different evaluators on the summarization task.(a) SERL vs. Qwen3-32B.(b) SERL vs. the base model Qwen3-8B.

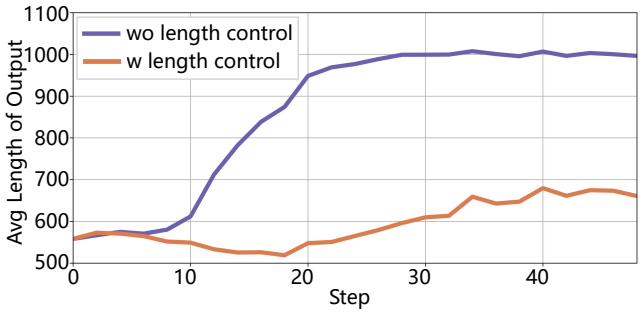

Figure 5: Comparison of average output length changes between the complete method and the method without length control mechanism during summarization training.

to assess the outcomes. Specifically, in the summarization task, we utilized GPT-4 Turbo and GPT-4o as evaluator models to evaluate the results between the model trained by our method and Qwen3-32B, as well as between the model trained by our method and the base model Qwen3-8B. As shown in Figure 4, the distributions of pairwise comparison judgments from different evaluator models are highly similar, indicating that the evaluation approach based on LLMs exhibits strong consistency.

4.6 Ablations and Analysis

We constructed four ablation variants, each removing a single component of the full framework: (1) Reward for Judge, (2) Reward for Actor, (3) Position Bias Mitigation Mechanism, and (4) Length Control Mechanism. The results of four experiments are shown in Table 5 and Figure 5.

Reward for Judge ($\mathcal{R}_J$) We remove the $\mathcal{R}_J$ component and keep the $\mathcal{R}_A$ component. The ablation variant without $\mathcal{R}_J$ consistently underperforms the complete method in three tasks. This confirms that strengthening the Judge's evaluation capability improves the Actor's generation performance, likely by providing more accurate feedback signals during training.

Reward for Actor ($\mathcal{R}_A$) We remove the $\mathcal{R}_A$ component and retain the $\mathcal{R}_J$ component. The ablation variant without $\mathcal{R}_A$ consistently underperforms the complete method across all three tasks, with the largest performance degradation observed among all ablation variants. This is because this variant only improves the model's evaluation ability, without directly enhancing the generation ability that correlates with the win rate.

Position Bias Mitigation Mechanism (PBMM) After removing the PBMM component described in Section 3.2, we observe that the Judge develops a preference for generations that appear in the second position. This positional bias undermines the model's evaluation ability and consequently hinders further improvements in generation ability.

Length Control Mechanism (LCM) As shown in Figure 5, for the summarization task, removal of LCM leads to a rapid increase in the average length, surpassing 1,000 tokens within a few iterations. In contrast, with LCM enabled, the average length remains stable around 600 tokens. LCM effectively controls length growth, enabling fair comparisons under minimal length discrepancies.

5 Conclusion

In this paper, we address the challenges of applying RL to improve the capabilities of LLMs on open-domain tasks. We propose SERL, an on-policy self-examining training framework. SERL introduces two synergistic reward mechanisms that rewards derived from Copeland-style pairwise comparison judgments without external signals to optimize model's generation ability and self-consistency rewards to optimize model's evaluation ability. Ultimately, this leads to improved performance of the model on open-domain tasks and outperforms existing self-improvement training methods. Additional material is provided in the extended version (Ou et al. 2025).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; et al. 2022b. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Chen, W.; Yin, M.; Ku, M.; Lu, P.; Wan, Y.; Ma, X.; Xu, J.; Wang, X.; and Xia, T. 2023. Theoremqa: A theorem-driven question answering dataset. *arXiv preprint arXiv:2305.12524*.

Copeland, A. H. 1951. A "Reasonable" Social Welfare Function. Mimeographed manuscript, University of Michigan, Department of Mathematics. Often cited as the original source for the Copeland method.

Dubois, Y.; Galambosi, B.; Liang, P.; and Hashimoto, T. B. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Dwork, C.; Kumar, R.; Naor, M.; and Sivakumar, D. 2001. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web*, 613–622.

Ethayarajh, K.; Xu, W.; Muennighoff, N.; Jurafsky, D.; and Kiela, D. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Fan, A.; Lewis, M.; and Dauphin, Y. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Gao, L.; Schulman, J.; and Hilton, J. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, 10835–10866. PMLR.

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

Hermann, K. M.; Kocisky, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Hu, J.; Zhang, Y.; Han, Q.; Jiang, D.; Zhang, X.; and Shum, H.-Y. 2025. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*.

Jaech, A.; Kalai, A.; Lerer, A.; Richardson, A.; El-Kishky, A.; Low, A.; Helyar, A.; Madry, A.; Beutel, A.; Carney, A.; et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Lanchantin, J.; Chen, A.; Lan, J.; Li, X.; Saha, S.; Wang, T.; Xu, J.; Yu, P.; Yuan, W.; Weston, J. E.; et al. 2025. Bridging Offline and Online Reinforcement Learning for LLMs. *arXiv preprint arXiv:2506.21495*.

Le, H.; Wang, Y.; Gotmare, A. D.; Savarese, S.; and Hoi, S. C. H. 2022. Coderl: Mastering code generation through pretrained models and deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 21314–21328.

Lee, H.; Phatale, S.; Mansoor, H.; Lu, K. R.; Mesnard, T.; Ferret, J.; Bishop, C.; Hall, E.; Carbune, V.; and Rastogi, A. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback.

Li, P.; Skripkin, M.; Zubrey, A.; Kuznetsov, A.; and Oseledets, I. 2025. Confidence Is All You Need: Few-Shot RL Fine-Tuning of Language Models. *arXiv preprint arXiv:2506.06395*.

Lin, C.-Y.; and Hovy, E. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 human language technology conference of the North American chapter of the association for computational linguistics*, 150–157.

Liu, J.; Zhu, Y.; Xiao, K.; Fu, Q.; Han, X.; Yang, W.; and Ye, D. 2023. Rltf: Reinforcement learning from unit test feedback. *arXiv preprint arXiv:2307.04349*.

Luo, M.; Tan, S.; Huang, R.; Patel, A.; Ariyak, A.; Wu, Q.; Shi, X.; Xin, R.; Cai, C.; Weber, M.; et al. 2025. Deepcoder: A fully open-source 14b coder at o3-mini level. *Notion Blog*.

Ma, X.; Liu, Q.; Jiang, D.; Zhang, G.; Ma, Z.; and Chen, W. 2025. General-reasoner: Advancing llm reasoning across all domains. *arXiv preprint arXiv:2505.14652*.

Ou, W.; Zheng, Y.; Sun, S.; Zhang, W.; Dong, B.; Zhu, H.; Huang, R.; Yu, G.; Yan, P.; and Qiao, Y. 2025. SERL: Self-Examining Reinforcement Learning on Open-Domain. arXiv:2511.07922.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744.

Park, R.; Rafailov, R.; Ermon, S.; and Finn, C. 2024. Disentangling length from quality in direct preference optimization. *arXiv preprint arXiv:2403.19159*.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36: 53728–53741.

Rein, D.; Hou, B. L.; Stickland, A. C.; Petty, J.; Pang, R. Y.; Dirani, J.; Michael, J.; and Bowman, S. R. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Shi, L.; Ma, C.; Liang, W.; Ma, W.; and Vosoughi, S. 2024. Judging the judges: A systematic investigation of position bias in pairwise comparative assessments by llms.

Shojaee, P.; Jain, A.; Tipirneni, S.; and Reddy, C. K. 2023. Execution-based code generation using deep reinforcement learning. *arXiv preprint arXiv:2301.13816*.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. F. 2020. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33: 3008–3021.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wang, Y.; Ma, X.; Zhang, G.; Ni, Y.; Chandra, A.; Guo, S.; Ren, W.; Arulraj, A.; He, X.; Jiang, Z.; et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290.

Wu, T.; Yuan, W.; Golovneva, O.; Xu, J.; Tian, Y.; Jiao, J.; Weston, J.; and Sukhbaatar, S. 2024. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. *arXiv preprint arXiv:2407.19594*.

Ye, D.; Ren, S.; Xu, H.; et al. 2023. UltraFeedback: Boosting Language Models with High-Quality Feedback. *arXiv preprint arXiv:2310.01377*.

Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; Yue, Y.; Dai, W.; Fan, T.; Liu, G.; Liu, L.; et al. 2025a. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Yu, T.; Ji, B.; Wang, S.; Yao, S.; Wang, Z.; Cui, G.; Yuan, L.; Ding, N.; Yao, Y.; Liu, Z.; et al. 2025b. RLPR: Extrapolating RLVR to General Domains without Verifiers. *arXiv preprint arXiv:2506.18254*.

Yuan, W.; Pang, R. Y.; Cho, K.; Sukhbaatar, S.; Xu, J.; and Weston, J. 2024. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 3.

Yue, Y.; Yuan, Y.; Yu, Q.; Zuo, X.; Zhu, R.; Xu, W.; Chen, J.; Wang, C.; Fan, T.; Du, Z.; et al. 2025. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Brooks, D.; Gonzalez, J. E.; et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Zhou, X.; Liu, Z.; Sims, A.; Wang, H.; Pang, T.; Li, C.; Wang, L.; Lin, M.; and Du, C. 2025. Reinforcing General Reasoning without Verifiers. *arXiv preprint arXiv:2505.21493*.